

Treball de Fi de Grau

Grau d'Enginyeria en Tecnologies Industrials

**Monitorización a través de IoT y análisis de datos de un
nodo de telecomunicaciones**

MEMÒRIA

Autor: Maria Font Salat

Director: Lluís Solano

Convocatòria: Gener 2018



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resumen

Este proyecto está basado en la creación de una plataforma cuyo objetivo final es monitorear y analizar parámetros tanto físicos como eléctricos en un nodo de telecomunicaciones. Este nodo es el encargado de dar conexión a internet a una zona limitada de la ciudad y es imprescindible su buen funcionamiento.

Para llevar a cabo todo eso, se procede a un diseño global de todos los aspectos, así que inicialmente se contemplará tanto la parte de *software* como la de *hardware*. Para definir correctamente estas dos partes, es necesario un estudio de los parámetros interesantes a monitorear y qué sensores son capaces de hacerlo. Seguidamente se caracterizarán el tipo de conexiones entre *hardware* y *software*, teniendo en cuenta que el objetivo final es poder controlar los sensores de forma remota.

En el momento en que se tengan los datos provenientes de los sensores, se creará un prototipo de algoritmo programado en lenguaje Python. Los datos se reciben en grandes cantidades, los cuales se acumulan en bases de datos y se deben exportar para realizar un buen tratamiento y análisis de datos. En este apartado se trabajará con herramientas como MongoDB, Robomongo y un editor Python con sus correspondientes librerías enfocadas al *datamining*.

El prototipo de plataforma debe servir de manual para futuras instalaciones, ya sea en la misma empresa en la que se ha desarrollado el proyecto como en otras del mismo ámbito con objetivos similares.

Juntamente con la planificación y los costos del proyecto, al final se dispone de un modelo que cumple con las especificaciones exigidas en un principio y que es capaz de dar solución a problemas tan importantes en una empresa dedicada a las nuevas tecnologías.

ÍNDICE

Resumen	2
ÍNDICE	3
LISTADO DE FIGURAS.....	5
LISTADO DE TABLAS	6
LISTADO DE ECUACIONES	6
GLOSARIO	7
PREFACIO	8
Origen y motivación del proyecto	8
1. INTRODUCCIÓN	9
1.1 Objetivos del proyecto	9
1.2 Estructura del proyecto	10
1.3 Alcance del proyecto	10
2. CONTEXTO SITUACIONAL.....	11
2.1 Descripción de la empresa	11
2.2 Situación actual	11
3. MARCO TEÓRICO	11
3.1 ¿Qué es un nodo de telecomunicaciones?.....	11
3.2 Posición geográfica de los nodos de Puntonet.....	12
3.3 Nodo Villaflora.....	13
3.3.1 Elementos nodo	13
3.4 Sistema de Monitoreo.....	14
3.5 Conectividad sensores	15
3.5.1 Ethernet, LAN y WLAN.....	15
3.5.2 Wifi y conectividad celular (2G, 3G y 4G)	16
4. DESCRIPCIÓN DE LA SOLUCIÓN.....	16
4.1 Qué parámetros hay que monitorear	17
4.1.1 Temperatura exterior	17
4.1.2 Humedad	18
4.1.3 Corriente que alimenta un circuito	19
4.1.4 Movimiento en el nodo	21
4.1.5 Temperatura interior.....	23
4.1.6 Consumo energía.....	25
4.2 Software	26
4.2.1 mFi Controller Software	26
4.3 Hardware.....	28
4.3.1 mPort.....	28
4.4 Exportación de data	29
4.4.1 MongoDB.....	30
4.4.2 Herramientas MongoDB.....	30
5. IMPLEMENTACIÓN DE LA SOLUCIÓN	30

5.1 Software	30
5.1.1 Instalación	30
5.2 Hardware.....	31
5.2.1 Conexión Ethernet.....	31
5.2.2 Conectar sensores	31
5.2.3 Pruebas previas	31
5.2.4 Conexión en el nodo.....	33
5.3 Exportar data	34
5.3.1 Mongo DB.....	34
5.3.2 Robomongo	34
5.3.3 Exportar a .CSV	36
5.3.4 Campos estudiados	36
6. PRUEBAS Y ANÁLISIS DE RESULTADOS	37
6.1 Test Data	37
6.1.1 Importancia Test Data	37
6.1.2 MOCKAROO	37
6.2 Algoritmo y análisis en Python	42
6.2.1 Python	42
6.2.2 Herramientas y librerías	43
6.2.3 Algoritmo.....	44
7. PLANIFICACIÓN Y COSTOS	51
7.1 Planificación temporal	51
7.2 Costos	52
8. IMPACTO SOCIAL Y AMBIENTAL	54
8.1 Impacto social.....	54
8.2 Impacto ambiental.....	54
CONCLUSIONES	55
AGRADECIMIENTOS.....	57
BIBLIOGRAFÍA	58
Referencias bibliográficas	58
Bibliografía complementaria.....	58

LISTADO DE FIGURAS

FIGURA 1: ESQUEMA GENERAL DEL PROYECTO	10
FIGURA 2: FUNCIONAMIENTO NODO TELECOMUNICACIONES.....	12
FIGURA 3: UBICACIÓN DEL NODO VILLAFLORA EN QUITO	13
FIGURA 4: FUNCIONAMIENTO SISTEMA DE MONITOREO	14
FIGURA 5: PASOS A SEGUIR PARA EL FUNCIONAMIENTO DE LA PLATAFORMA.....	15
FIGURA 6: IMAGEN DEL SENSOR AF10	17
FIGURA 7: ESPECIFICACIONES DEL SENSOR AF10.....	17
FIGURA 8: IMAGEN SENSOR HWG-STE	19
FIGURA 9: ESPECIFICACIONES SENSOR HWG-STE.....	19
FIGURA 10: ESPECIFICACIONES TÉCNICAS MFI-CS	20
FIGURA 10: FUNCIONAMIENTO MFI-CS	20
FIGURA 12: ESPECIFICACIONES TÉCNICAS DEL MFI-DS.....	22
FIGURA 13: ESPECIFICACIONES TÉCNICAS MFI-MSW	22
FIGURA 14: ESPECIFICACIONES TÉCNICAS SENSOR MFI-THS	24
FIGURA 15: ESPECIFICACIONES TÉCNICAS MPOWER PRO.....	26
FIGURA 16: PRECIO MPOWER PRO.....	26
FIGURA 17: REPRESENTACIÓN DEL NODO.....	27
FIGURA 18: INTERFAZ GRÁFICA DEL MFI CONTROLLER SOFTWARE.....	28
FIGURA 19: VISTA FRONTAL DEL MPORT	29
FIGURA 20: VISTA LATERAL DEL MPORT.....	29
FIGURA 21: ESQUEMA CONEXIÓN PRUEBA FINAL	33
FIGURA 22: ESQUEMA CONEXIÓN EN EL NODO.....	33
FIGURA 23: CONFIGURACIÓN FINAL DEL MPORT	34
FIGURA 24: CONFIGURACIÓN EN ROBO_MONGO PARA CONECTAR CON LA BASE DE DATOS	35
FIGURA 25: ESTADO INDICANDO QUE LA CONEXIÓN CON MONGODB ES CORRECTA	35
FIGURA 26: INTERFAZ DE LA BASE DE DATOS ACE	35
FIGURA 27: CREACIÓN DATASET TEMPERATURA EXTERIOR.....	38
FIGURA 28: CÓDIGO PARA SACAR LA MEDIA Y LA STD DEV	39
FIGURA 29: VALORES TEMPERATURAS EN VILLAFLORA.....	39
FIGURA 30: CREACIÓN DATASET TEMPERATURA INTERIOR	40
FIGURA 31: CREACIÓN DATASET HUMEDAD	40
FIGURA 32: CREACIÓN DATASET ENERGÍA ELÉCTRICA.....	41
FIGURA 33: CREACIÓN DATASET MOVIMIENTO.....	41
FIGURA 34: GRÁFICO POPULARIDAD DE LOS LENGUAJES DE PROGRAMACIÓN 2016	43
FIGURA 35: GRÁFICO VALORES TEMPERATURA INTERNA.....	45
FIGURA 36: CÓDIGO ALARMA TEMPERATURA INTERIOR	46
FIGURA 37: GRÁFICO QUE REPRESENTA EL SENSOR DE MOVIMIENTO	47
FIGURA 38: INTERFAZ RULES TAB CON LA ALARMA DEL SENSOR DE CORRIENTE (LÍMITE SUPERIOR)	48
FIGURA 39: INTERFAZ RULES TAB CON ALARMA DEL SENSOR DE CORRIENTE (LÍMITE INFERIOR)	49
FIGURA 40: GRÁFICO SCATTER DE LAS RELACIONES	50
FIGURA 40: DIAGRAMA DE GANTT PARTE 1	52
FIGURA 41: DIAGRAMA DE GANTT PARTE 2	52

LISTADO DE TABLAS

TABLA 1: DESCRIPCIÓN DE LOS ELEMENTOS ACTIVOS	14
TABLA 2: PRECIO SENSOR MFI-CS	21
TABLA 3: SENSORES DE MOVIMIENTO ESCOGIDOS	21
TABLA 4: PRECIOS SENSORES DE MOVIMIENTO	23
TABLA 5: : LÍMITES TEMPERATURA INTERIOR SEGÚN EQUIPO	24
TABLA 7: PRECIO SENSOR MFI-THS	25
TABLA 8: RESUMEN COSTOS PROYECTO	53

LISTADO DE ECUACIONES

ECUACIÓN 1: CÁLCULO UMBRALES CORRIENTE	20
ECUACIÓN 2: FÓRMULA COEFICIENTE CORRELACIÓN	49
ECUACIÓN 3: CÁLCULO COSTO INGENIERO JUNIOR	53
ECUACIÓN 4: CÁLCULO HORAS A DEDICAR AL TFG	53
ECUACIÓN 5 Y 6: CÁLCULO COSTO TUTORES	53

GLOSARIO

IP estática o dinámica (Internet Protocol): Tipo de IP's que existen. Se puede tener una IP estática, que será siempre la misma, o una IP dinámica, que varía según ciertas necesidades.

IoT (Internet of Things): es un concepto que se refiere a la interconexión digital de objetos cotidianos con Internet.

SAI (Sistema de Alimentación ininterrumpida): en inglés *uninterruptible power supply* (UPS), es un dispositivo que gracias a sus baterías u otros elementos almacenadores de energía, puede proporcionar energía eléctrica por un tiempo limitado y durante un apagón eléctrico a todos los dispositivos que tenga conectados.

AC (Altern Current): Corriente alterna

.csv (Comma Separated Values): Los archivos CSV son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas o punto y coma, y las filas por saltos de línea.

Big Data: es un concepto que hace referencia a un conjunto de datos tan grandes que aplicaciones informáticas tradicionales de procesamiento de datos no son suficientes para tratar con ellos.

Msnm: Metros Sobre el Nivel del Mar

PoE (Power over Ethernet): es una tecnología que incorpora alimentación eléctrica a una infraestructura LAN estándar. Permite que la alimentación eléctrica se suministre a un dispositivo de red (switch, punto de acceso, router, teléfono o cámara IP, etc.) usando el mismo cable que se utiliza para la conexión de red.

DHCP (Protocolo de configuración dinámica de Host): es un protocolo cliente-servidor que proporciona automáticamente un host de protocolo Internet (IP) con su dirección IP y otra información de configuración relacionados como, por ejemplo, la puerta de enlace predeterminada y la máscara de subred. El servidor DHCP mantiene un grupo de direcciones IP y concede una dirección a cualquier cliente DHCP cuando se inicia en la red.

PREFACIO

Origen y motivación del proyecto

Para dar con la idea final del proyecto se juntaron diferentes motivaciones: primero de todo, el interés desarrollado por las nuevas tecnologías junto con las asignaturas de electrónica e informática. Posiblemente una parte de eso se debe a que vemos muchos aspectos de estas asignaturas aplicados a la vida real, donde uno se da cuenta de la importancia que tiene tener ni que sea una noción sobre conexiones, datos guardados en la nube, qué hacer cuando dispones de datos masivos, etc.

Eso se juntó con el auge del Internet de las Cosas o *Internet of Things* (IoT). La idea de conectar todas las cosas que nos rodean a una misma nube, para que a partir de allí se puedan tomar decisiones que nos faciliten la vida a todos, es una idea que ya de principio llama la atención. Esas cosas de las que hablamos, pueden ser cosas relacionadas con cualquier sector que, en este caso, es el sector de las telecomunicaciones.

Pero para que el IoT sirva de algo y como se ha dicho, ayude y facilite la vida de la gente, hay que hacer algo al respecto, y eso tiene que ver con el Big Data y el análisis de estos datos. El Big Data es otro de los temas de los que habla todo el mundo actualmente, ya que cada segundo se generan millones de datos procediendo de cualquier cosa con los que mucha gente no sabe qué hacer. Es por eso que saber analizar estos datos, tratarlos correctamente y sacar conclusiones es algo realmente interesante y útil.

Entonces se decidió que juntando las cosas mencionadas, se podía llegar a tener un proyecto interesante, aplicado a la vida real y que podía servir de ayuda a muchas empresas. Pero como todo proyecto, no basta con tener la idea final de lo que se quiere obtener, sino que el proceso para llegar a ella es igual de importante. Y es en este punto donde se acabó de definir el proyecto al completo.

1. INTRODUCCIÓN

En las empresas de Telecomunicaciones representa un factor de alta prioridad el tener control sobre todos los posibles aspectos que afecten a la presentación de los servicios, concretamente teniendo en cuenta que esta clase de servicios requieren de una disponibilidad 24/7, todos los días del año. El motivo es que en caso de presentarse problemas se aplicarán multas que representan altos costos para las empresas e inconformidad para los clientes finales, cosa que perjudica gravemente a la empresa proveedora de estos servicios.

Teniendo en cuenta la distribución geográfica y las condiciones que se dan en la actualidad para los nodos de telecomunicaciones, considerando que son un punto crítico de control y acceso para el personal de mantenimiento ya que una gran mayoría de los nodos se encuentran ubicados en cerros y montañas de difícil acceso físico, y según la experiencia para el caso de la empresa donde se lleva a cabo el proyecto, el tener conocimiento previo de los eventos que ocurren en el nodo representa una herramienta muy útil y apropiada.

Es por eso que cada vez más las empresas que se encuentran con problemas de este tipo, se plantean sistemas que les faciliten la vida a la vez de solucionarles los problemas. Además, teniendo en cuenta el *boom* del *Big Data* que hay actualmente, las empresas no quieren quedarse atrás y no dudan en innovar.

No se puede hablar de sensores y *Big Data* sin mencionar el *Internet of Things* (IoT). El IoT consiste básicamente en conectar todos los objetos que nos rodean a la nube. Muchos expertos prevén que en unos años, toda la sociedad estará controlada a través de sensores, los cuáles recogerán millones y millones de datos y gracias al análisis de estos, se podrá analizar y prever el comportamiento de prácticamente todo. Este proyecto está relacionado en cierto modo a todo este mundo, ya que cumple muchos de los requisitos para ser un proyecto relacionado con el IoT; sensores, conexión remota y análisis de datos.

1.1 Objetivos del proyecto

Implementar un sistema prueba de monitoreo de ciertos parámetros tanto eléctricos como físicos que proporcione la información adecuada para enfrentar, estudiar y llegar a solucionar los problemas que se vienen presentado.

El proyecto a desarrollar está focalizado en el diseño de la plataforma partiendo de cero, lo que implica las siguientes tareas:

- Estudio de qué parámetros son importantes a la hora de monitorizar y con qué sensor se controlarán
- Instalación e implementación del software utilizado
- Instalación e implementación del hardware
- Extracción de datos de los sensores
- Diseño del estudio y tratamiento de datos

1.2 Estructura del proyecto

Para cumplir los objetivos mencionados, hay que enfocar el proyecto juntando el ámbito de la gestión de proyectos y el de las telecomunicaciones e informática.

El porqué de enfocarlo desde el punto de vista de la gestión de proyectos es básico. Se parte de una idea bastante genérica, donde se sabe el punto de partida y lo que se desea obtener. Pero entremedio hay muchos aspectos que hay que gestionar, organizar, ordenar y definir.

Todo el trayecto desde el punto de partida hasta el punto final son en general, tareas de tipo informático o de telecomunicaciones, así que no basta sólo con saber gestionar el proyecto, sino también saber de qué trata.

La plataforma tiene diferentes partes esenciales pero muy diferentes entre ellas:

El mPort, que se verá posteriormente que es la base a la cuál van conectados los sensores, se conecta inalámbricamente a un router del nodo y envía información a su software, que una vez conectado correctamente, permitirá extraer los datos en el formato escogido para así poderlos tratar. El análisis de datos se hará a través de un lenguaje de programación y se estudiarán las relaciones e interacciones destacables.

En la *Figura 1* siguiente se muestra el esquema global del proyecto donde se ven diferenciadas las diferentes partes:

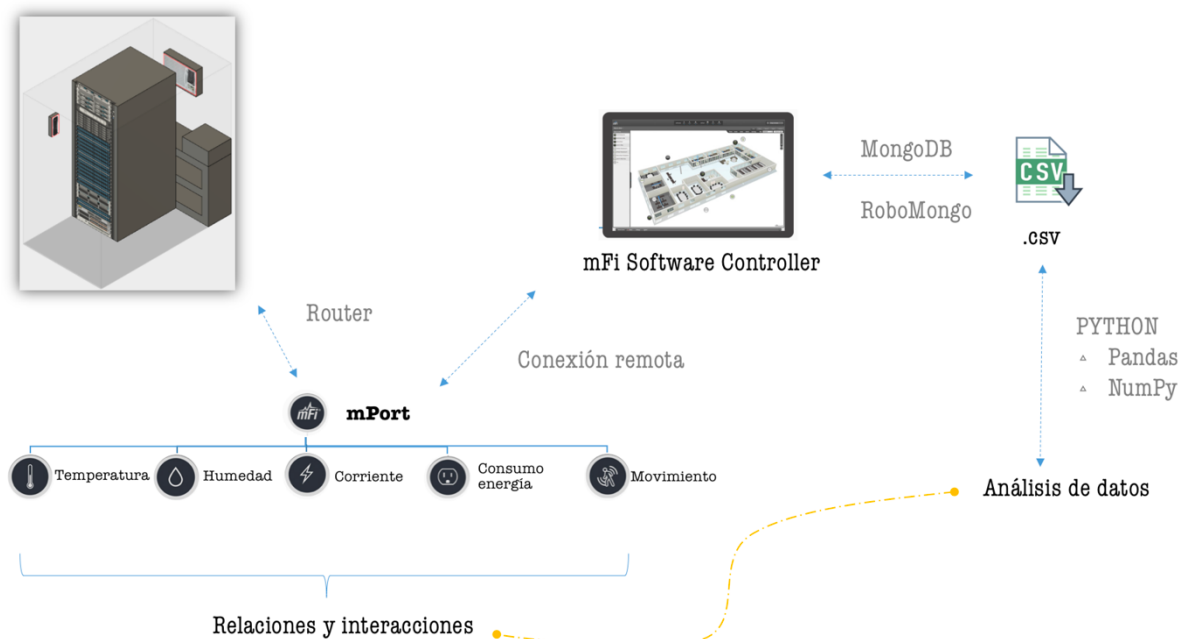


Figura 1: Esquema general del proyecto

1.3 Alcance del proyecto

Este trabajo pretende ser un prototipo y manual de un futuro proyecto a implementar. En primer lugar, servirá de herramienta para la instalación de una plataforma de monitoreo de parámetros en los nodos de telecomunicaciones.

Por otro lado, al ser un prototipo, significa que en gran parte no se trabajará con datos reales, pero que estará totalmente preparado para hacerlo.

Es un proyecto muy interesante y muy amplio que en el apartado de análisis de datos, se podría seguir y desarrollar una herramienta que permitiera no sólo ver cuál es el comportamiento actual de estos parámetros, sino estudiarlos de forma predictiva.

2. CONTEXTO SITUACIONAL

2.1 Descripción de la empresa

Puntonet es una empresa de telecomunicaciones líder en Ecuador. Tiene diferentes oficinas y diferentes departamentos, cada uno de ellos separados dependiendo de si tratan temas corporativos o particulares. Se encargan principalmente de dar conexión internet a nivel de banda ancha y fibra óptica, pero actualmente también disponen de un equipo técnico encargado del Data Center más grande del país y encargado de tratar los temas de Cloud. Eso incluye ser responsables desde los equipos físicos que hay en la misma oficina de trabajo, hasta de todo el mantenimiento y funcionamiento correcto de los servidores.

Es en este equipo de Data Center y Cloud donde se desarrolló la totalidad del proyecto.

2.2 Situación actual

Para solucionar el problema planteado, es importante analizar bien la situación actual de la empresa donde se desarrolla el proyecto, para ver cuáles son las necesidades exactas, las condiciones y facilidades de las que se dispone.

El problema viene dado por la fallada de los servicios que proporciona la empresa y la forma actual de gestionarla es la siguiente: visita de los técnicos al nodo de forma “inmediata”. A falta de previsión de estos fallos, se llega tarde al nodo y el servidor ya ha fallado. Esta situación representa unos costos muy elevados debido a la inconformidad de los clientes y las multas que hay que pagar.

Como facilidades que la empresa dispone: la parte del proyecto de la conexión remota entre nodo y oficina está relativamente solventada. La posibilidad de utilizar la misma red de transporte de la empresa representa una muy eficiente opción para el proyecto, ya que la comunicación con el sitio remoto era una de las grandes limitantes que se presentaba.

3. MARCO TEÓRICO

3.1 ¿Qué es un nodo de telecomunicaciones?

En informática y en telecomunicación, de forma muy general, un nodo es un punto de intersección, conexión o unión de varios elementos que confluyen en el mismo lugar.

Las principales funciones del nodo estudiado para este proyecto son:

- Dar conexión Wifi
- Dar conexión de fibra óptica

Y el funcionamiento es el que se puede observar en la *Figura 2*:



Figura 2: Funcionamiento nodo telecomunicaciones

3.2 Posición geográfica de los nodos de Puntonet

Teniendo en cuenta que cada nodo puede dar conexión a la zona con la que tiene una línea de visión directa, la red de nodos de la empresa es muy amplia ya que Quito, ciudad dónde se lleva a cabo el proyecto, queda definida por las pendientes de los volcanes que la rodean y dos turones justo en medio de la ciudad.

A pesar de que la idea sería implementar esta plataforma en todos los nodos, con todo proyecto hay que empezar poco a poco, para así ser capaces de valorar la viabilidad del proyecto y hacer una inversión razonable. Es por eso que el proyecto se centrará en el nodo instalado en el barrio de Villaflora, en Quito, a unos 2800 msnm.

El motivo de la elección de este nodo se debe a las facilidades que nos presenta la zona donde está situado, ya que el nodo está instalado encima de otras oficinas de la misma empresa y permite acceder a él sin necesidad de contactar con los propietarios del edificio que acostumbran a ser edificios privados.



Figura 3: Ubicación del nodo Villaflora en Quito. Fuente: Google Maps

3.3 Nodo Villaflora

3.3.1 Elementos nodo

A partir de un informe proporcionado por la empresa [1], se pueden clasificar los elementos que forman parte del nodo elegido:

Elementos pasivos

- Antenas
 - Parabólica
 - Panel
 - Sectoriales Frec. 2.4GHz y 5GHz
 - Sectorial 360°
 - Cables coaxiales

Elementos activos

Equipo	Tipo
Equipo Mikrotik (MKT) 411AH con radio XR5 (Bakhaul)	Router
Equipo MKT 433AH con radio XR2	Router
Equipo MKT 433AH con radio XR5	Router
Equipo MKT 433AH con radio R52n	Router
Equipo Omnitik	Punto de acceso
PoEs 24V (1A)	Power over Ethernet
Fuentes regulables 3A	
Switch administrable	Switch

Tabla 1: Descripción de los elementos activos

3.4 Sistema de Monitoreo

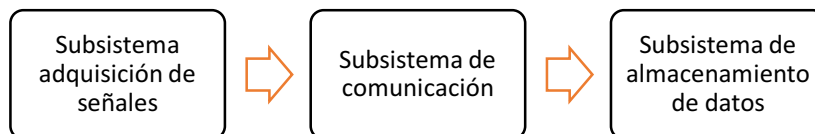


Figura 4: Funcionamiento sistema de monitoreo

- Subsistema adquisición de señales: es el sistema formado por los sensores y encargado de tomar señales externas.
- Subsistema de comunicación: es el encargado de transportar los datos a través de la red para ser entregados al subsistema de almacenamiento. Consiste básicamente en la red de transporte de la empresa.
- Subsistema de almacenamiento de datos: encargado de realizar la manipulación de los datos enviados desde el subsistema de adquisición de señales y realizar la visualización correspondiente de los diferentes eventos que se generan.

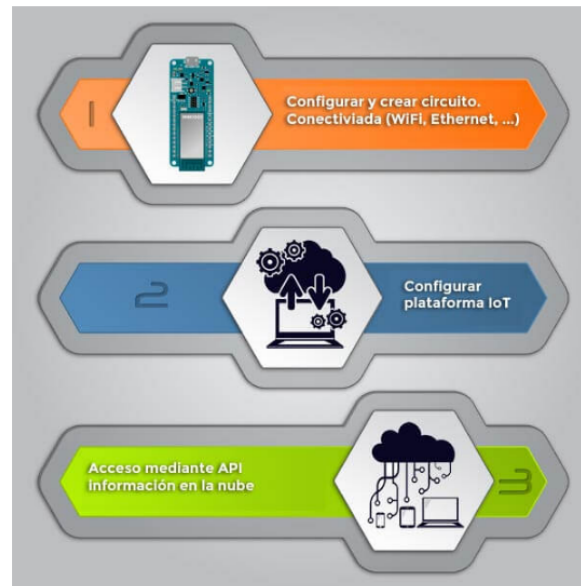


Figura 5: Pasos a seguir para el funcionamiento de la plataforma

3.5 Conectividad sensores

Una vez los sensores están instalados, es imprescindible pensar en que tipo de conexión usaremos para conectar estos con nuestra plataforma de gestión de datos.

Las conexiones se centran básicamente en dos grandes grupos:

3.5.1 Ethernet, LAN y WLAN

LAN (*Local Area Network*), hace referencia a una red que interconecta una o más computadoras dentro de una geográfica limitada, a través de hardware y software, con el fin de compartir archivos, recursos y hasta una conexión a Internet.

Las LAN convencionales utilizan típicamente par trenzado, cables coaxiales o en algunos casos, fibra óptica (ancho de banda infinito).

Detrás de esta denominación se encuentra un estándar llamado Ethernet, el cual determina las particularidades físicas y eléctricas que debe poseer una red tendida con este sistema (todos los elementos que tienen papel dentro de una red y como deben ser conectados entre ellos).

También llamamos Ethernet a una red LAN conectada a través de cable, en el cual se deben interconectar entre si todos los dispositivos que la conforman mediante un dispositivo llamado router.

Cuando se conecta a través de ondas de radio de alta frecuencia, pasa a llamarse WLAN (*Wireless Local Area Network*). Esta nos permite prescindir de cables e instalaciones complicadas. Es una red de computadoras a distancia de unas pocas decenas o cientos de metros. Potencialmente, la transmisión no es tan rápida como la proporcionada por una LAN convencional, pero para la mayoría de usuarios, esa velocidad no es una limitación.

A partir de aquí, el elemento más importante de cualquier red mencionada es el cable Ethernet, cable utilizado para interconectar todos los dispositivos que conforman una LAN. Es el encargado de llevar todos los datos que usualmente transitan por una red.

3.5.2 Wifi y conectividad celular (2G, 3G y 4G)

WI-FI es la abreviatura de Wireless Fidelity. El término es en la actualidad un nombre de marca utilizado para marcar productos que pertenecen a una categoría de dispositivos WLAN. Es un nombre de marca registrada para los dispositivos de la marca que cumplen con los estándares IEEE 802.11. Un dispositivo WI-FI ready simplemente significa que está listo para el funcionamiento de la red dentro de una WLAN.

Su función es básicamente permitir a un dispositivo electrónico el intercambio de datos o conexión a Internet, sea a 2,4GHz o 5GHz.

Esta conexión puede ser menos segura que las conexiones por cable (Ethernet), ya que el atacante no necesita estar físicamente conectado. Pero como ventaja, las redes Wifi permiten el acceso de múltiples computadoras, a diferencia de Ethernet, que está limitada a los puertos instalados y a los usuarios registrados.

Hablando ahora de la conectividad móvil, se pueden observar mejoras muy rápidas y constantes en ella. Hasta la fecha han surgido cuatro generaciones de tecnología móvil. La primera denominada 1G, sólo permitía realizar llamadas de voz. En la década de los 90, apareció la segunda generación, denominada 2G o GSM, que ya admitía la transmisión de datos a baja velocidad. Con el siglo XXI apareció 3G, que logró alcanzar velocidades de hasta 40 Mbps. Actualmente las necesidades han hecho que nazca la cuarta generación, 4G (LTE), que ha multiplicado por 5 la velocidad de su predecesora. Una de las increíbles cosas que se puede llegar a hacer con la tecnología 4G es almacenar y sincronizar en la nube todo tipo de archivos.

Aunque esta tecnología estaba orientada a la telefonía móvil, desde hace unos años las operadoras de telefonía móvil ofrecen servicios exclusivos de conexión a Internet mediante módem USB, sin necesidad de adquirir un teléfono móvil, por lo que cualquier computadora puede disponer de acceso a Internet.

Tanto Wifi como las otras redes comentadas, consumen una gran cantidad de energía, pero son ampliamente soportadas y con gran cobertura. Poco a poco, van apareciendo redes totalmente adecuadas para tratar con el IoT. Son redes que consumen muy poco y eso es perfecto para poder conectar sensores y que puedan funcionar mucho tiempo.

El motivo por el que no se trabajó con estas redes en el proyecto es porque, como se ha dicho, están apareciendo ahora y en Ecuador no hay cobertura todavía.

4. DESCRIPCIÓN DE LA SOLUCIÓN

Se implementará todo el diseño de la plataforma siguiendo un orden concreto, empezando por la definición de los parámetros que hay que monitorear junto con la propuesta del sensor que se encargará de eso, una breve explicación del software utilizado, introducción al hardware y finalmente, la descripción de las herramientas que permitirán la extracción de datos.

4.1 Qué parámetros hay que monitorear

4.1.1 Temperatura exterior

■ Descripción del parámetro

Se decide estudiar este parámetro que, aunque sea una medida sobre la cual no podamos efectuar ningún cambio, será interesante relacionarla más adelante con los valores de la temperatura de dentro del *shelter* (llamada así la parte que contiene los equipos dentro en el nodo), para ver cómo afecta la temperatura exterior con la temperatura interior.

■ Límites del parámetro

No hay límites ni alertas sobre este parámetro ya que sólo sirve de estudio.

■ Qué tipo de sensor

En el proyecto actual de la empresa no consta ninguna intención de instalar un sensor para medir la temperatura exterior porque actualmente hay muchas estaciones meteorológicas que te permiten sacar estos datos.

Como propuesta, se podría plantear la opción de instalar un sensor que tome estos datos sin necesidad de recurrir a empresas externas.

Posible propuesta: AF00, AF10, and AF20 OUTSIDE TEMPERATURE SENSORS, de la marca Honeywell [2].



Figura 6: Imagen del sensor AF10. Fuente: Datasheet Honeywell

SPECIFICATION

Nominal value	
Pt 1000	1000 Ω at 0 °C (32 °F)
NTC 10k	10 k Ω at 25 °C (77 °F)
NTC 20k	20 k Ω at 25 °C (77 °F)
Accuracy	
Pt 1000 (IEC751 Class B)	± 0.3 °C at 0 °C (32 °F)
NTC 10k	± 0.2 °C at 25 °C (77 °F)
NTC 20k	± 0.2 °C at 25 °C (77 °F)
Sensitivity	
Pt 1000	≈ 3.85 Ω / K
NTC 10k	-440 Ω / K at 25 °C (non-linear)
NTC 20k	≈ -934.5 Ω / K at 25 °C (non-linear)
Connection	two-wire (2x0.22 mm ²)
Time constant	≤ 3 min.
Enclosure material	polycarbonate, UL94-V0 approved
Protection class	IP54 / IP65 as per EN 60529
Ambient Limits (wiring box)	
Storage temperature	-30...+70 °C (-22...+158 °F)
Humidity	0...95% rh, non-condensing
Dimensions	see Fig. 1 on page 2

Figura 7: Especificaciones del sensor AF10

Permite la medida de la temperatura exterior gracias a su carcasa resistente al agua.

Es adecuado también porque para el monitoreo se pueden usar los diferentes sistemas de Control:

- Eagle
- Lion
- Tiger
- Excel 5000, 500 y 800
- Excel Web y Web II
- Módulos Honeywell
- ...

- El precio de este dispositivo a través de Amazon sería de: AF00 (35\$), AF10 (120\$), AF20 (137\$)

4.1.2 Humedad

■ Descripción del parámetro

La humedad es un parámetro muy importante que influye directamente con el funcionamiento de los equipos del nodo.

Cabe destacar que en un país como Ecuador, donde puedes encontrar climas muy diferentes, éste parámetro cobra importancia. Hay nodos instalados a 4000msnm en zona andina igual que nodos a 0msnm con clima totalmente húmedo y costero. Es por eso que aunque nuestro proyecto se lleve a cabo en un sitio con pocos problemas de humedad, para una futura instalación nacional, será de gran uso.

■ Límites parámetro

Los sensores de humedad miden la humedad relativa de su entorno, y en un cuarto de telecomunicaciones con equipos electrónicos, es aconsejable mantener estos valores entre el 30% y el 55% [3].

Si se sobrepasan estos valores, se debe disparar una alerta para evaluar cómo se procede y si es necesario un cambio de aire en el shelter.

■ Tipo de sensor

Igual que en el caso interior, no hay intención ahora mismo de la empresa de implementar este tipo de sensor, ya que la marca con la que se trabaja habitualmente, Ubiquiti, no dispone de ningún sensor que se encargue de tomar estas medidas.

A pesar de eso, se hace una propuesta para una posible futura implementación debido a la importancia del parámetro.

Una posible opción de un sensor que mida la humedad interior: Termómetro ethernet HWg-STE [2].

El HWg-STE es un termómetro Ethernet que permite controlar temperatura y humedad por IP, es decir, controla temperatura y humedad remotamente. El HWg-STE envía alertas por email en caso de superar los límites definidos. Se suministra con una única sonda de temperatura y humedad, alimentador y software. A pesar de que la temperatura interior ya

se monitorea a través del sensor integrado en nuestra plataforma mFi (que se verá seguidamente), actualmente no se encuentran sensores sólo de humedad, así que se tomarán las medidas de humedad en este caso que son las que interesan.

El software HWg-PDMS registra los datos, muestra gráficos y permite la extracción de los datos a MS Excel.



Figura 8: Imagen sensor HWg-STE. Datasheet Ditecom

Parámetros

Ethernet	100 Mbit
Protocolos IP	ARP, TCP/IP (HTTP,SMTP) UDP/IP (SNMP)
DHCL para asignación IP	✓
SNMP	Puerto 161 Ver 1.0 / 2.0 (en parte)
Protocolo M2M	SNMP, XML (sobre http)
Precisión	$\pm 0,5^{\circ}\text{C}$ (-10°C a $+85^{\circ}\text{C}$) $\pm 2^{\circ}\text{C}$ (-55°C a $+125^{\circ}\text{C}$)
Alimentación	5V / 1W
Temperatura de funcionamiento	-10°C a 60°C

Figura 9: Especificaciones sensor HWg-STE

- Precio: 155€ + IVA

4.1.3 Corriente que alimenta un circuito

■ Descripción parámetro

El nodo estudiado es un caso de red monofásica. Las líneas trifásicas se van desdoblando en monofásicas para alimentar a pequeños consumidores como es el caso de nuestro nodo. Eso significa que está formado por una única corriente alterna o fase y por lo tanto todo el voltaje varía de la misma forma.

Así pues, estudiaremos el corriente de la fase que proviene de la energía pública y por otro lado, el corriente que va directo al SAI.

■ Límites del parámetro

El corriente nominal que está pasando por el cable que da corriente al SAI es de unos 10 mA.

Para calcular los umbrales, se hace a través de los VA del SAI¹, los 110 V que se suministran al nodo y un factor de potencia (FP) de 0.8.

$$3000 \text{ VA} \times 0.8 = 2400 \text{ VA}$$

$$\frac{2400 \text{ VA}}{110 \text{ V}} = 21.8 \text{ A}$$

Ecuación 1: Cálculo umbrales corriente

Enviar alerta cuando el corriente disminuya por debajo del 70% del valor calculado, es decir, un 70% de 21.8 A que equivale a unos **15 A**. Como vemos, este valor se aleja mucho del nominal, y eso se debe a que estos equipos son capaces de soportar altos valores de corriente antes de que sea peligroso.

En referencia al valor del umbral inferior, se estudió que a partir de unos **7 mA** debería dispararse una alarma.

■ Qué tipo de sensor

El equipo escogido es el mFi Current Sensor (mFi-CS) de la marca Ubiquiti **¡Error! No se encuentra el origen de la referencia..**

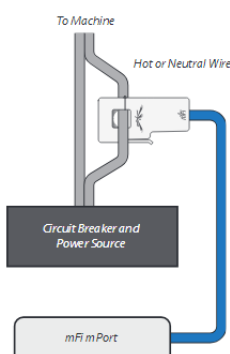


Figura 11: Funcionamiento mFi-CS. Fuente: mFi Datasheet

mFi-CS	
Dimensions	32 x 57 x 22 mm (1.26 x 2.24 x 0.87")
Size of Opening	13 x 13 mm (0.51 x 0.51")
Weight	50 g (1.76 oz)
Ports	(1) mFi RJ45 Port
Input Current	0 to 100A
Load	Linear*
Core Material	Ferrite
Bobbin	Nylon
Mechanical Strength	Number of Switching Operations ≥ 1000 Times (Tested at 20° C)
Dielectric Strength (Between Shell and Output)	6000VAC/1 min.
Fire Resistance Rating	UL94-V0
Operating Temperature	-25 to 70° C (-13 to 158° F)
Operating Humidity	≤85% Noncondensing

* Non-linear loads are approximated as linear loads (sine wave).

Figura 10: Especificaciones técnicas mFi-CS

¹ La potencia en vatios es la potencia real consumida por el equipo. Se denomina Voltios-Amperios a la "potencia aparente" del equipo, y es el producto de la tensión aplicada y la corriente que por él circula.

- **Funcionamiento sensor:**
El sensor mide corriente AC en el circuito, no potencia (potencia es voltaje multiplicado por intensidad). El sensor mide el valor eficaz del voltaje y lo convierte a corriente (usando $1V = 100A$).
- **Precio: (2 sensores)**

Precio (\$)	Precio + Impuestos (\$)	Total (\$)
11,32	14,26	28,53

Tabla 2: Precio sensor mFi-CS

4.1.4 Movimiento en el nodo

■ Descripción parámetro

El objetivo será detectar movimiento, lo que significa monitorear tanto la apertura de puertas de los nodos o centros de datos como el movimiento en su interior. Es importante principalmente para las dos situaciones descritas a continuación:

- ✓ *Caso1:* Robo
- ✓ *Caso2:* Visita de un técnico. En este caso es para controlar cuando un técnico de la empresa acude al nodo. Es útil para situaciones en que se ha modificado alguna cosa dentro del nodo y después no funciona o simplemente para poder ponerse en contacto con el técnico encargado de dicha modificación.

■ Límites del parámetro

Como no es muy común que haya movimiento en los nodos, de momento es suficiente con crear una alarma en el momento que se detecte algo. Así se podrá contar las veces que entra algún técnico y comprobar que coincidan con las previstas.

Para automatizar más el proceso, una idea interesante sería que se disparase la alerta solo en caso de apertura de puerta/movimiento en horas/días atípicos. También encender la cámara en el momento de apertura de puerta para así poder registrar que está pasando.

■ Qué tipo de sensor

¡Error! No se encuentra el origen de la referencia.

Equipo	Descripción
mFi-DS	mFi, Door Sensor
mFi-MSW	mFi, Wall Mount Motion Sensor

Tabla 3: Sensores de movimiento escogidos

mFi-DS	
Dimensions	50 x 9.5 x 7.7 mm (1.97 x 0.37 x 0.30")
Weight (Magnet and Switch)	350 g (12.35 oz)
Contact Form	Form A (SPST)
Maximum Rating	1.0A @ 28VDC
Initial Contact Resistance	0.3 Ω Maximum
Operating Range	20 mm (Typical)
Wiring	Screw Terminal
Mounting	Adhesive or Screws (Included)
Operating Temperature	-10 to 60° C (14 to 140° F)
Operating Humidity	≤80% Noncondensing

Figura 12: Especificaciones técnicas del mFi-DS

mFi-MSW	
Dimensions	146 x 66 x 52 mm
Weight	127 g
Power Supply	9 - 16VDC, 12VDC Typical
Current Drain	35 mA @ 12VDC
Infrared Sensor	Dual Element
Microwave Sensor	DRO, Patch Antenna
Alarm Period	3 Seconds
Alarm Output	NC/NO, 24VDC, 0.2A Max.
Detection Range	10 x 10 m, 110° @ 25° C
Mounting	Wall/Ceiling (Kits included)
Mounting Height	2.3 m Typical
Detection Speed	0.3 to 3.0 m/sec
RFI Immunity	Avg. 10 V/m (80 to 2,000 MHz)
Operating Temperature	-10° to 45° C
Humidity	95% RH Max.
Ports	(1) mFi RJ45 Port

Figura 13: Especificaciones técnicas mFi-MSW

○ **Funcionamiento sensor:**

Los dos sensores funcionan con un voltaje de 12 V que reciben del mPort. También reciben 24 V del pin Output. Envían una señal de vuelta al pin Input si no se detecta ningún movimiento. En el caso de detectarse, el sensor abre el circuito con el fin de que no se reciba la señal. Eso es lo que indica que se ha detectado movimiento. Es un sensor de movimiento de doble tecnología, utilizando tanto los sensores de movimiento infrarrojos pasivos y microondas para minimizar la probabilidad de falsas detecciones.

- Precio:

Equipo	Precio (\$)	Precio + Impuestos (\$)	Total (\$)
mFi-DS	35,09	44,21	44,21
mFi-MSW	24,9	31,37	31,37

Tabla 4: Precios sensores de movimiento

4.1.5 Temperatura interior

■ Descripción parámetro

Controlar la temperatura en el interior del *shelter* es una de las cosas más importantes debido a la alta escala de integración y complejidad presentada en los diversos equipos de telecomunicaciones. Estos generan temperaturas elevadas por su funcionamiento, así que es vital mantenerlos a una temperatura adecuada.

No sólo es el calentamiento de los propios equipos, sino también las condiciones meteorológicas. Aunque el *shelter* disponga de un buen aislamiento exterior, la temperatura no es la misma en horas de sol que en ausencia de él.

En nuestro caso, la temperatura recomendada está entre los 0 y 40 °C, la misma que se toma del sensor colocado directamente en contacto con los equipos en el nodo de telecomunicaciones.

Aunque sean temperaturas extremas, eso se debe a que la situación geográfica de los nodos en Ecuador es muy diversa. En la costa los nodos se encuentran a nivel de mar, con clima húmedo y temperaturas altas, mientras que en la zona más andina los nodos están situados a altitudes que a veces superan los 3000 msnm, con temperaturas por debajo de los 10°C.

■ Límites del parámetro

Equipos activos

Equipo	T máx (°C)	T mín (°C)
Equipo Mikrotik (MKT) 411AH con radio XR5 (Bakhaul)	70	-40
Equipo MKT 433AH con radio XR2	70	-40
Equipo MKT 433AH con radio XR5	70	-40
Equipo MKT 433AH con radio R52n	70	-40
Equipo Omnitik	70	-30

Switches:		
- Cisco Catalyst 9300 Series Switches	45	-5
- Cisco Catalyst 9400 Series Switches	45 (menos de 3000msnm) o 40 (más de 3000msnm)	-5
- Cisco Catalyst 9500 Series Switches	40	0

Tabla 5: : Límites temperatura interior según equipo

▪ Qué tipo de sensor

El sensor escogido es el sensor mFi-THS de la marca Ubiquiti **¡Error! No se encuentra el origen de la referencia..**

Specifications

mFi Temperature Sensor	
Dimensions	100 x 85 x 27.8 mm
Weight	80 g
Temperature Range	-10 to 50 °C
Accuracy at +25 °C, 50% RH	±0.5 °C ±3% RH
Temperature Accuracy from -10 to +50 °C	Max. ±1.3 °C
Response Time	<15 Seconds
Port	(1) mFi RJ45 Port
Operating Temperature	-20 to 60 °C
Operating Humidity	0 to 95% Noncondensing

Figura 14: Especificaciones técnicas sensor mFi-THS

El rango de temperaturas que el sensor puede indicar está dentro del rango de operación de los equipos activos.

También se cumplen los límites en el sentido que el sensor puede trabajar desde -20 °C hasta 60 °C.

○ Funcionamiento:

Recibe 12V del mPort. El voltaje del sensor es leído por los pins del mPort. Da el resultado en Celsius.

- Precio: (2 sensores)

Precio	Precio + Impuestos	Total
19,24	24,24	48,48

Tabla 6: Precio sensor mFi-THS

4.1.6 Consumo energía

■ Descripción parámetro

Controlar la energía consumida, tanto para asegurar un buen rendimiento todo el rato como para hacer un consumo eficiente de esta.

Dentro del nodo, la energía es consumida por los equipos activos que hay en este y por las baterías que alimentan los SAI. Estos últimos actúan como sistema de respaldo y es importante que les llegue alimentación en caso de necesitarlo.

Posibles problemas con la energía:

- Eventos transitorios (rayos, tormentas eléctricas... que conducen al aumento repentino de tensión y de corriente).
- Corte energía
- Pico de voltaje
- Subvoltaje
- Sobretensión
- Ruido de Línea
- Variación de frecuencia

■ Límites del parámetro

Los valores de voltaje adecuados están alrededor de los 110-125 VAC (voltios de corriente alterna). Si se sobrepasan, será señal de que algo no está funcionando bien.

■ Qué tipo de sensor

A parte de ser una regleta de alimentación, su objetivo es también monitorear cada puerto AC (8) y determinar el consumo de energía por cada uno.

El equipo escogido es el mPower Pro de la marca Ubiquiti [¡Error! No se encuentra el origen de la referencia..

mPower PRO	
Dimensions	250.37 x 116.7 x 42 mm
Weight	754.4 g
Mounting	Wall (Kit Included)
Power	110-125VAC, 50-60 Hz, 15 A Per Outlet, 15 A Total
Outlets	8
Networking Interface	(1) 10/100 Ethernet Port
Wi-Fi Standards	802.11b/g/n
Memory	16 MB RAM, 8 MB Flash
LEDs	1 LED (Status)
Operating Temperature	0 to 40° C
Humidity	10 to 95% RH Max.

Figura 15: Especificaciones técnicas mPower Pro

- Precio:

Precio (\$)	Precio + Impuestos (\$)	Total (\$)
81,52	102,72	102,72

Figura 16: Precio mPower Pro

4.2 Software

4.2.1 mFi Controller Software

mFi es un software de gestión M2M diseñado para funcionar con los productos de mFi Ubiquiti y es el escogido para gestionar toda esta plataforma. Desde mFi es posible controlar los productos que hayan sido instalados y conectados adecuadamente, ver las medidas que se toman en tiempo real, crear un sistema de alertas indicando qué condiciones harán saltar estas alertas, ubicar a un mapa la ubicación de tus sensores, etc. El mismo software no dispone de la opción de exportar datos fuera de la plataforma, pero se verá más adelante la alternativa seguida para este proceso.

Interfaz gráfica

Map Tab

El software te permite subir cualquier mapa personalizado para una visualización real de la ubicación de tus productos.

El mapa utilizado para representar el nodo se realizó con el programa Fusion360 de versión gratis.

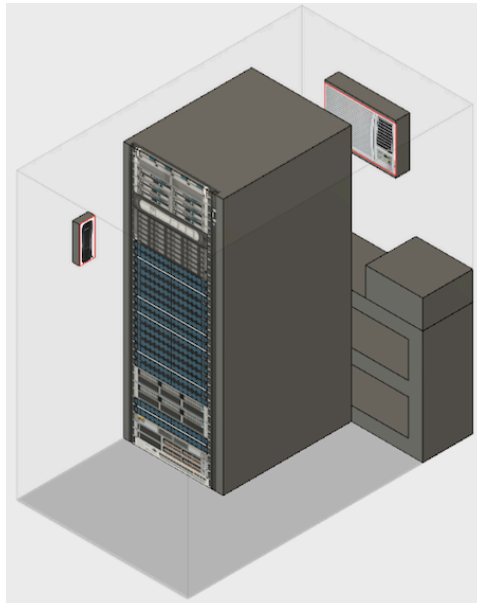


Figura 17: Representación del nodo. Fuente: Fusion360

Data Tab

El apartado de Data Tab te permite ver los datos de cada dispositivo. Puedes organizarlo y visualizarlo por tiempo, dispositivo, tipo, etc. También te permite comparar datos de diferentes sensores.

Events Tab

Muestra toda la información relativa a los eventos que ocurren, pudiendo filtrar los mismos por fecha, sensor, tipo de evento, etc.

También tiene la opción de visualización a través de gráficos.

Scenes Tab

Permite definir acciones (URL y Script) genéricas para luego activarlas desde el Rules Tab.

Rules Tab

En el Rules Tab, se pueden diseñar y crear ciertas acciones, que se activaran cuando se cumplan unas condiciones indicadas.

Aquí es donde se puede escoger que tipo de notificación recibir si algún parámetro sale de sus valores normales. Los tipos de notificación son los siguientes:

- Email: manda un correo a la dirección de correo especificada
- Output
- URL Post: Escribir un post en la dirección URL especificada
- Enable/Disable Rule: activar o desactivar otra regla al activarse la misma
- Run Scene: si anteriormente se ha definido un escenario en concreto se puede activar y así no hay que generar una nueva acción

Energy Tab

Se controla la energía consumida por las regletas así como el costo de este consumo, ya que previamente se deben indicar los \$/ kWh.

Device Tab

Simplemente muestra una lista de los productos, tanto mPorts como sensores que hay conectados y su estado actual.

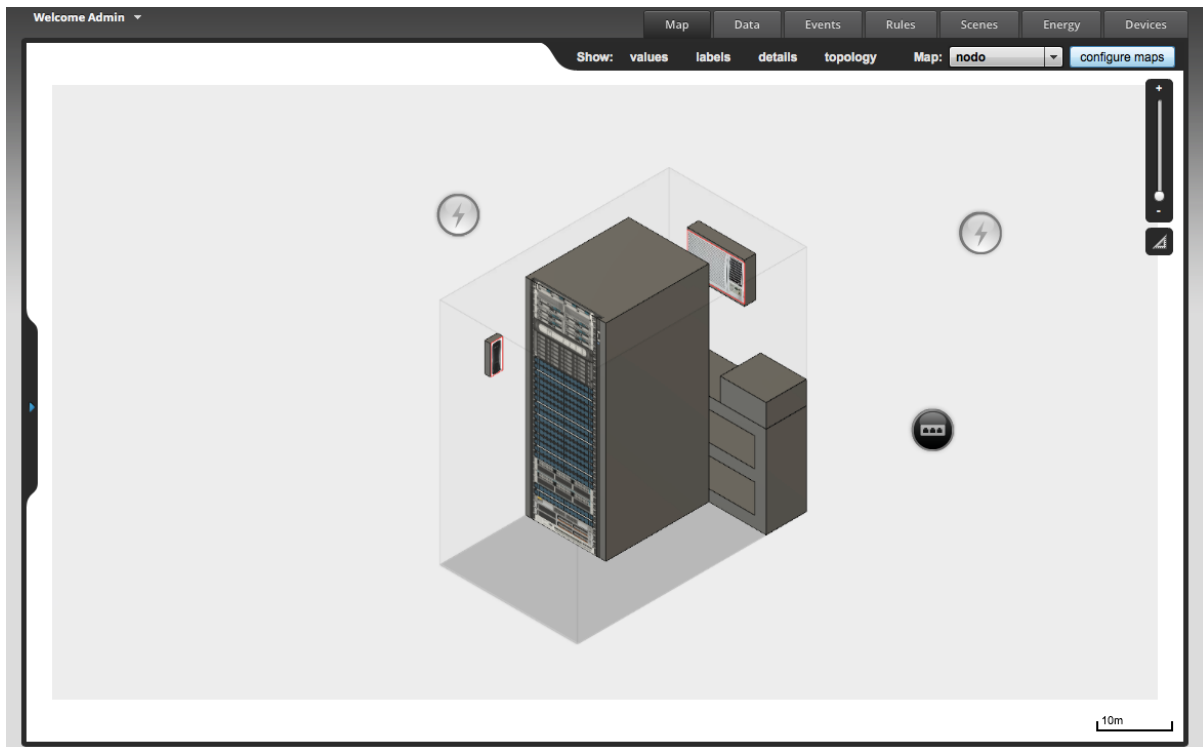


Figura 18: Interfaz gráfica del mFi Controller Software

4.3 Hardware

El hardware lo componen los sensores y el equipo encargado de ser un punto de conexión entre sensor y demás dispositivos (el llamado mPort).

Hay que considerar también un subgrupo de hardware formado por los routers, switches y cables utilizados.

4.3.1 mPort

El mPort es el que se encargará de realizar las conexiones sensor-Ethernet y se podría considerar la base de la plataforma.

A simple vista, en la parte superior izquierda, se puede apreciar el estado de las tres posibles conexiones, tal como indica la Figura 19:

- Power LED: Emitirá luz verde cuando el dispositivo esté correctamente conectado a una fuente de energía.
- Ethernet LED: Verde cuando haya una conexión activa Ethernet, y emitirá un *flash* cuando haya actividad.
- Status LED: Amarilla cuando es encendida por primera vez por defecto. Verde cuando el mPort esté correctamente integrado a la red y funcione adecuadamente. Y finalmente emitirá *flash* cuando desde el mFi software se active la tecla *Locate* (su función es localizar el sensor).

Es importante saber que a cada mPort sólo se pueden conectar dos sensores.

Hardware Overview

Front View

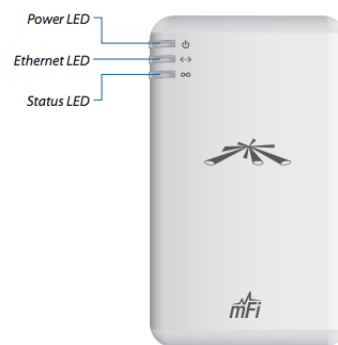


Figura 19: Vista frontal del mPort. Fuente: mPort Quick Start Guide

Side View

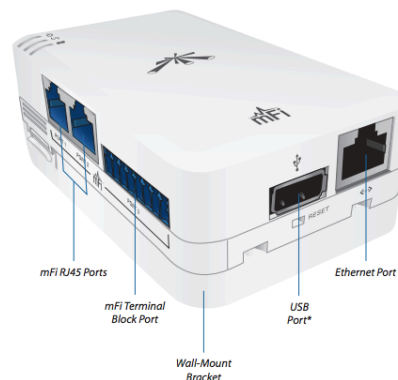


Figura 20: Vista lateral del mPort. Fuente: mPort Quick Start Guide

4.4 Exportación de data

Para proseguir al estudio de los datos que recibimos de los sensores, es imprescindible exportar los datos que se muestran en el apartado de Data Tab en el software y hay que descargarlos para poderlos analizar bien.

Los datos son almacenados en una base de datos MongoDB en la misma computadora. Se puede consultar esta base usando diferentes herramientas DB.

4.4.1 MongoDB

MongoDB es una base de datos orientada a documentos. Esto quiere decir que en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, representación lineal de JSON.

Una de las diferencias más importantes con respecto a las bases de datos relacionales, es que no es necesario seguir un esquema.

MongoDB está escrito en C++.

Cualquier aplicación que necesite almacenar datos semi estructurados puede usar MongoDB.

4.4.2 Herramientas MongoDB

Hay diferentes herramientas que permitirán acceder a la base de datos y leerla. Entre ellas se encuentran MongoVUE, Pentaho o Robomongo. A pesar de que la más utilizada puede que sea MongoVUE, sólo está disponible para Windows, así que queda descartada.

Robomongo es la más fácil de usar y que se adapta bien al sistema MAC OS X.

Robomongo

Robomongo es una herramienta multiplataforma con la que se puede administrar gráficamente nuestras bases de datos. Actualmente se ha convertido en un importante aliado para los usuarios de MongoDB.

Esta herramienta no se basa en un sistema Web, sino que se instala como un cliente en nuestro equipo. Una vez instalado, permite conectarse al servidor de base de datos de forma sencilla y ver los datos que está dando el mFi Software Controller.

5. IMPLEMENTACIÓN DE LA SOLUCIÓN

5.1 Software

5.1.1 Instalación

UNIFY SOFTWARE CONTROLLER

Requerimientos del sistema:

- Microsoft Windows 7, Windows Vista, Windows XP, Mac OS o Ubuntu Linux versión 11.04
- Sistema operativo recomendado de 64-Bits
- 2 GB de memoria RAM
- Java Runtime Environment 1.6

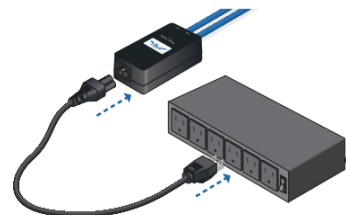
Requerimientos del navegador web:

- Mozilla Firefox, Google Chrome, Microsoft Internet Explorer 8, Safari
- FlashPlayer 10

5.2 Hardware

5.2.1 Conexión Ethernet

1. Conectar un cable Ethernet al puerto Ethernet
2. Conectar el otro extremo del cable Ethernet al puerto Ethernet llamado POE del adaptador POE
3. Conectar un cable Ethernet de tu propia LAN al puerto Ethernet del adaptador POE llamado LAN
4. Conectar el cable de alimentación al puerto correspondiente en el adaptador PoE. El otro extremo del cable de alimentación va conectado a la regleta de alimentación



5.2.2 Conectar sensores

Cada mPort tiene un menú de configuración y detalles al cual se puede acceder fácilmente a través de su icono. Allí se podrá añadir los sensores conectados al correspondiente mPort, indicando su nombre y el tipo de sensor que es. Si el cable desde el mPort hasta el sensor está correctamente conectado, el sensor está listo para empezar a medir.

5.2.3 Pruebas previas

Antes de llevar a cabo la instalación de los sensores en el nodo correspondiente, se realizan unas pruebas en oficina para asegurar que todos los pasos previos a la visualización de datos se cumplen.

El primer paso es conectar el mPort a la plataforma *software*, para lo que se utiliza un TP-LINK 150Mbps Wireless and router creando así una red a la que se conectará el ordenador (nombre red: TP-LINK-B142AO).

Para configurar el router, se conecta un cable Ethernet extremo1 al router, extremo2 al pc Windows. Allí se cambia la IP a la que viene por defecto del TP-LINK (conexiones de red, conexión de área local) y seguidamente, desde el navegador se accede a <http://tplinklogin.net> (web de configuración).

Cambiar nombre de la red si se desea, contraseña, etc. y permitir el servidor DHCP (dará direcciones IP a los diferentes equipos que se conecten a la red (IP: 192.168.48.200)).

Configuración mPort

Para que el mFi Controller Software detecte y reconozca nuestros dispositivos mPort, hay que configurarlos de forma correcta.

Por defecto, vienen con una IP asignada a la cual se puede acceder y allí configurar a que plataforma mFi van a estar vinculados, junto con el tipo de conexión *Wireless* a la que se van a conectar. Es aquí donde hay que seleccionar la red TP-LINK-B142AO. Es entonces cuando se verá un cambio en la luz del mPort que indica que están correctamente integrados en la red.

Para asegurar que el ordenador con el que se trabaja detecta este mPort, se puede descargar el programa Ubiquiti Discovery Tools, desde dónde aparecerá el dispositivo con sus características.

Cabe destacar que, si un mismo mPort se conecta a plataformas diferentes del mFi Controller, hay que reiniciar el dispositivo para que la otra plataforma lo detecte.

Instalación en MacBook (vía Wifi)

El primer paso será llevar a cabo la instalación de la plataforma para poder acceder desde un portátil, un MacBook en este caso. Se realiza la conexión vía Wifi a través de la red mencionada anteriormente y se asigna una dirección IP que está en la misma subred (IP: 192.168.48.202).

Una vez completado este proceso, conectar el ordenador desde dónde se trabajara (MacBook) a la red TP-LINK-B142AO (con dirección IP: 192.168.48.202).

Se abre el mFi Software Controller (misma dirección IP que TP-LINK-B142AO) donde ya automáticamente aparecerá el mPort listo para añadirlo.

El mismo software le asigna a cada mPort una dirección IP diferente, pero dentro de la subred de trabajo. En este caso, el primer mPort tiene una dirección 192.168.48.102, desde donde ahora se podrán cambiar sus características.

Instalación PC (vía cable)

Hay que recordar que en el PC se trabaja a través de una máquina virtual con sistema operativo Windows 7, así que hay que asegurar que ambas están conectadas a la misma subred; en este caso IP 192.168.48.201 para el ordenador normal, y IP 192.168.48.204 para la máquina virtual. Como se observa, todas son direcciones que asigna el router en el rango permitido que se le dio.

Como es obvio, y una vez descargado también el software para esta máquina, el mFi Controller se abrirá con una IP igual que la que había en el MacBook, ya que será la misma plataforma (IP: 192.168.48.202). Como se adoptó el mPort anteriormente, se ve la misma pantalla simultáneamente en los dos ordenadores.

Conexión de prueba final

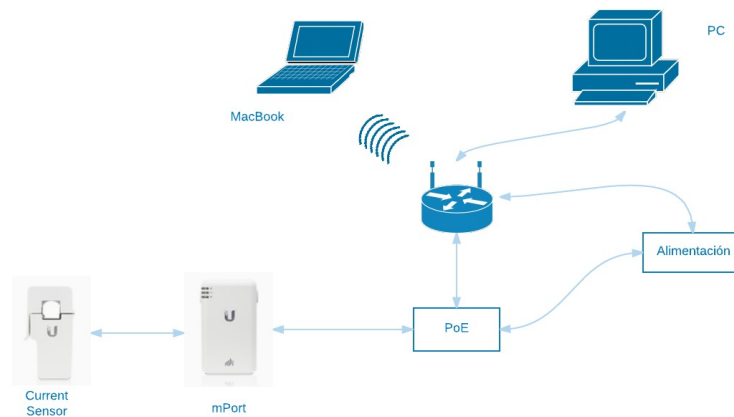


Figura 21: Esquema conexión prueba final

5.2.4 Conexión en el nodo

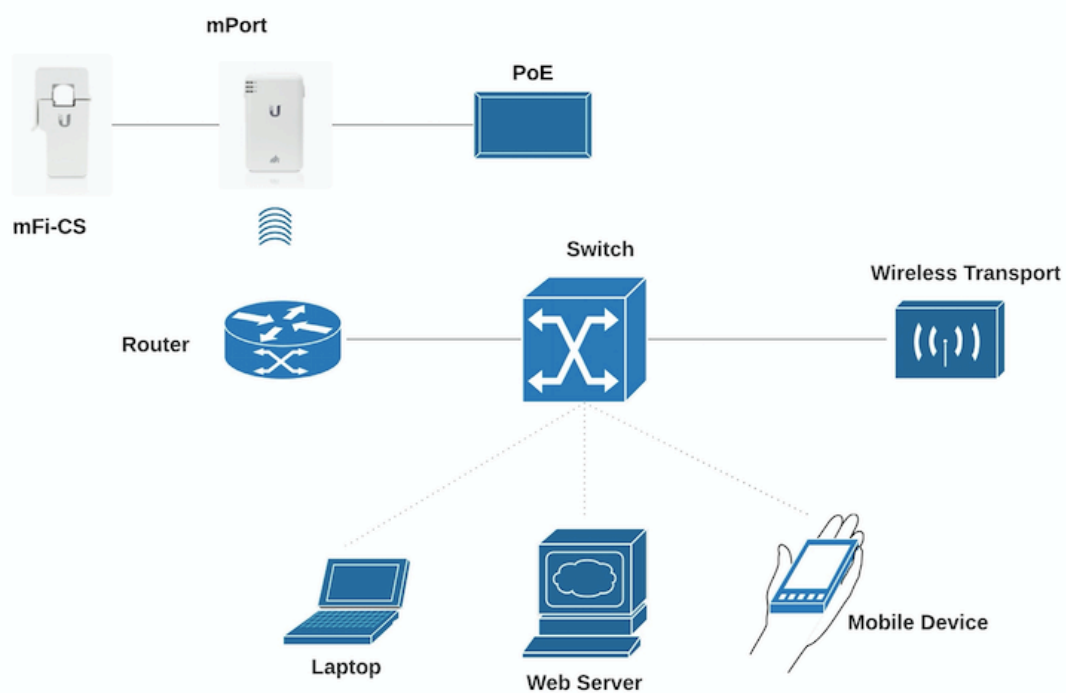


Figura 22: Esquema conexión en el nodo

El esquema anterior muestra el funcionamiento de la red mFi. En el mismo nodo hay un router con IP estática para que se pueda acceder desde cualquier dispositivo que no se encuentre cerca. Este router está conectado a un switch que se encuentra en el *shelter*.

El mPort accede inalámbricamente a esta red, permitiendo la integración de este en el mFi Controller Software. Es importante cambiar la IP del software en el que estará integrado el mPort ya que ahora es una estática con IP diferente. También importante añadir en la configuración del mPort a qué red inalámbrica se conectará.

Introduciendo la dirección IP anterior se podrá acceder remotamente a los datos que están mandando los sensores.

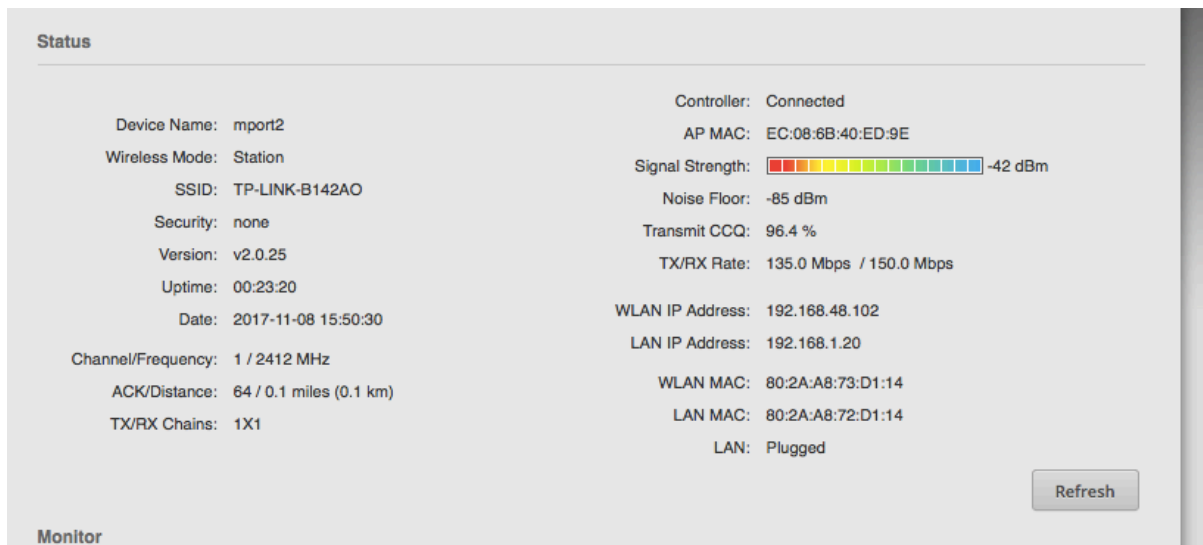


Figura 23: Configuración final del mPort

5.3 Exportar data

5.3.1 Mongo DB

Para poder acceder a la base de datos de MongoDB simplemente hay que seguir el proceso de descargar MongoDB y desde la terminal del ordenador, llamarlo para crear el acceso al puerto deseado.

5.3.2 Robomongo

Instalación

Una vez descargado, se debe crear una nueva conexión que permita conectarse a dónde se está trabajando con el *software* mFi. A través de comentarios de otros usuarios de la plataforma, se sabe que el *host* a cuál debemos conectarnos es el *localhost* del mismo ordenador, y que el *port* corresponde al 37017 por defecto (Figura 24).

De esta forma, se puede acceder a la base de datos de MongoDB que contiene los datos de los sensores en la plataforma mFi Software Controller.

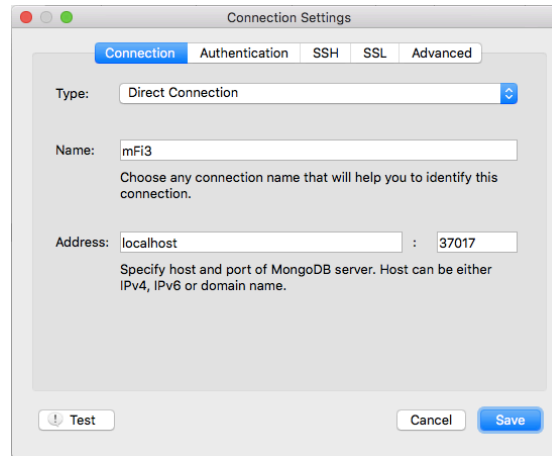


Figura 24: Configuración en RoboMongo para conectar con la base de datos

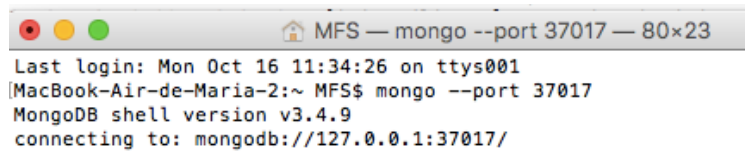


Figura 25: Estado indicando que la conexión con MongoDB es correcta

Una vez la conexión se ha realizado con éxito y se tiene acceso a los datasets del localhost, es posible ver en *Collections* la base de datos que nos interesa llamada “Ace”, mostrada en la Figura 26.

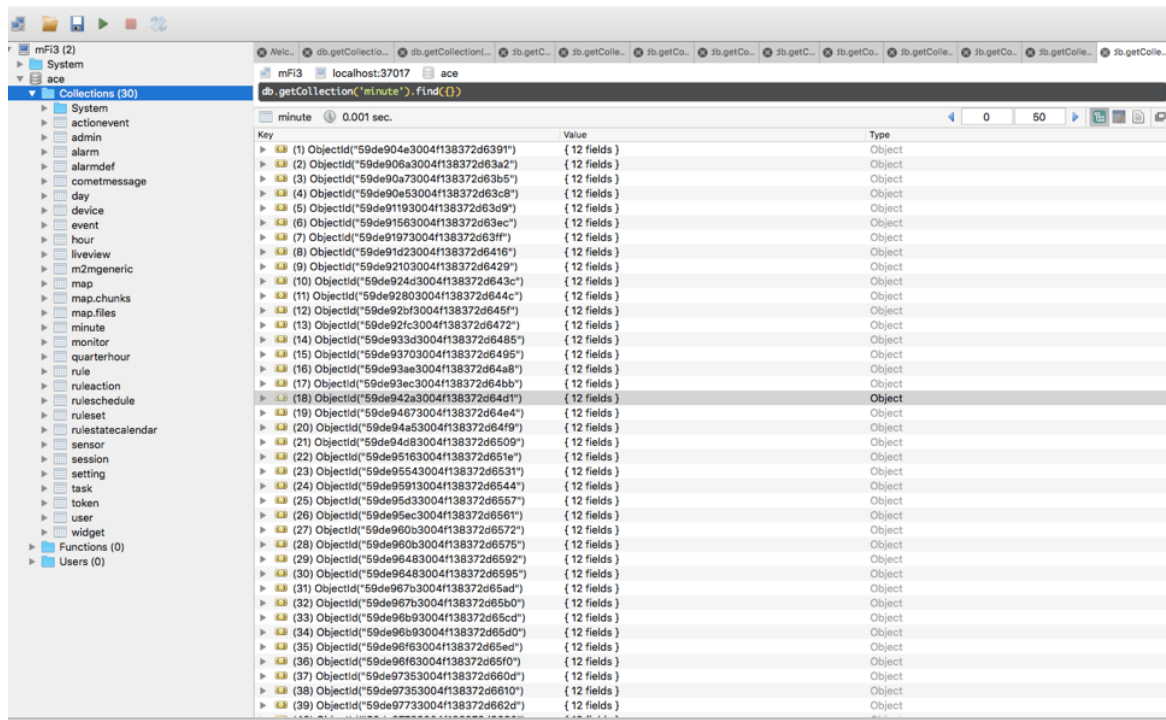


Figura 26: Interfaz de la base de datos ACE

Hay muchas colecciones y cada una de ella contiene ciertos datos sobre el *software*. Entre ellos, los más destacados son: los sensores que hay, los mapas subidos a la plataforma, las alarmas creadas, las veces que una alerta se disparó, etc. y finalmente, los datos sobre las medidas de los sensores.

5.3.3 Exportar a .CSV

Hasta aquí es muy interesante pero se necesita exportar los datos a un formato que permita analizarlos y estudiarlos, como es .csv (*Comma Separated Values*). Los ficheros CSV son un tipo de documento simple para representar datos en forma de tabla, donde las columnas se separan por comas y las filas por saltos de línea.

Se puede hacer a través del comando en la terminal del ordenador:

- Comando genérico:

```
mongoexport --port 37017 --db nombre base de datos --collection nombre
coleccion --fields nombre de los fields que se desean exportar --
type=csv --out nombre archivo.csv
```

- Ejemplo concreto:

```
mongoexport --port 37017 --db ace --collection minute --fields
timee,sId,max,min,ave,last_val,last_val_time --type=csv --out
mFimminute4.csv
```

5.3.4 Campos estudiados

Colección: MINUTE

- Time: hora en la que empieza el intervalo de toma de medida
- sId: da la identificación del sensor que es. La da con un formato largo y difícil de identificar, pero se cambia una vez en el editor Python para que sea más fácil de leer.
ObjectId("59de90453004f138372d638f") = Sensor1
ObjectId("59de95e33004f138372d655d") = Sensor2
- max: máximo valor registrado en el intervalo de tiempo
- min: mínimo valor registrado en el intervalo de tiempo
- ave: valor medio registrado en el intervalo de tiempo
- last_val: último valor registrado en el intervalo de tiempo
- last_val_time: instante del último valor registrado

Los campos que dan tiempo, lo dan en formato EPOCH (o también llamado Tiempo Unix o Tiempo POSIX).

Este sistema es usado para la descripción de instantes de tiempo: se define como la cantidad de segundos transcurridos desde la medianoche UTC del 1 de enero de 1970. Es universalmente usado en muchos sistemas computacionales e incrementa cada segundo, sin realizar cálculos necesarios para determinar el año, la hora y el minuto, haciendo ilegible la fecha para humanos.

Es por eso que existe la necesidad de convertir este tiempo al formato legible para humanos, y a través de Python y de la librería Datetime es posible:

```
df1['field']=df1['field']/1000  
df1['field'] = df1['field'].apply(lambda x:  
datetime.datetime.fromtimestamp(x))
```

Colección: EVENT

- Time: instante en el que se genera el evento. También en formato EPOCH
- Rulename: nombre de la alerta
- Sensor: en qué sensor se esta dando el evento
- Sensor_value: qué valor de corriente se está dando cuando se dispara la alerta

Como en esta colección también se muestran los eventos así como la conexión y desconexión de eventos, los cuales no interesan para el estudio, se procede a eliminar todas las filas que muestran valores NaN, que son justamente las que dan alertas de este tipo que no interesa.

6. PRUEBAS Y ANÁLISIS DE RESULTADOS

6.1 Test Data

6.1.1 Importancia Test Data

Cualquier aplicación que se esté desarrollando o cualquier plataforma que se esté preparando requiere de su testeo bajo condiciones lo más próximas a la realidad posible. Cuando la aplicación o la plataforma esté en uso, contará con muchísimos usuarios que harán que tengas que tratar con millones de datos acumulados. Es por eso que es esencial poder probar con una cantidad parecida de datos antes de llevar el proyecto a la realidad.

Como no es posible obtener datos reales de la mayoría de sensores que se prevén instalar en un futuro, se decide llevar a cabo el estudio con datos generados aleatoriamente a través de una plataforma especializada en eso.

El motivo por el cuál no es posible obtener datos tiene que ver con problemas relacionados con la exportación de productos desde el extranjero a Ecuador. Estrictos controles, largos trayectos y dificultades legales impidieron la llegada de los sensores a tiempo.

Pero aun así, se definieron los umbrales de trabajo de cada parámetro y se crearon una serie de datasets lo más cercanos a los futuros datasets reales.

6.1.2 MOCKAROO

¿Qué es?

Mockaroo es una gran herramienta para los testers. Se trata de una herramienta web desarrollada por Mark Brocato, desde la que vamos a poder generar hasta 100.000 líneas de datos realistas para pruebas que podremos exportar en formato CSV, Tab-Delimited, JSON, SQL, Excel y DBUnit XML.

Tener ficheros csv con datos de prueba realistas es de una inestimable ayuda. Y tenerlos con la rapidez y calidad que da mockaroo es todavía mejor. Además, probar con datos realistas hará que las pruebas de la aplicación sean más robustas, puesto que será posible detectar errores que podrían producirse con datos reales.

Hasta ahora, una posible opción era generar estos archivos con herramientas como Microsoft Excel u Open Office Calc, con las que copiando y arrastrando se podían generar archivos interesantes en un tiempo aceptable.

Pero Mockaroo va más allá. Permite indicar los campos que requiere el archivo deseado; tanto el nombre, como el tipo de campo que se quiere. Hay infinidad de tipos de campos, no desde un punto de vista de desarrollo (string, integer, boolean, ...) sino desde un tipo de vista funcional: nombre, apellido, ciudad, tarjeta de crédito (diferenciando entre visa, mastercard o american express), código de país, color, talla, hora, teléfono, tipo MIME,...

Creación de datasets

Temperatura exterior

Definición de valores y umbrales

Temperatura ext Save Changes

Field Name	Type	Options
row	Row Number	blank: 0 % <i>fx</i> ×
last_val_time	Dataset Column	Current Sensor <i>time</i> sequential blank: 0 % <i>fx</i> ×
ave	Normal Distribution	mean: 14,23 std. dev: 5,29 decimals: 2 blank: 5 % <i>fx</i> ×

Figura 27: Creación dataset temperatura exterior

Como se ha comentado, se quiere aproximar el dataset a la realidad en la medida de lo posible. En el apartado A.I del anexo se pueden observar que formato tendrán los datos reales, es por eso que se crean los campos con el nombre que serán creados directamente en el mFi Controller Software.

La columna de *last_val_time* corresponderá al valor del tiempo de toma de medida del parámetro (columna *time* en el dataset real). Así mismo, la columna *ave* será el valor de la medida.

En referencia a la columna de tiempo que se genera, se decide que la opción más cómoda y viable es simplemente copiar la columna de tiempo del sensor de corriente (datos reales), así estaremos trabajando con los mismos instantes de tiempo y el análisis se acercará más a la realidad.

Para generar los valores de *ave* se tiene en cuenta la siguiente tabla de temperaturas (Figura 29) registradas en el barrio de Villaflora (ubicación dónde se encuentra el nodo) usando la generación aleatoria de valores siguiendo una distribución normal.

A través de una pequeña programación en Python y utilizando la librería NumPy que nos permite hacer básicas operaciones estadísticas, sacamos la media (*mean*) y la desviación estándar (*standard deviation*).

```
import numpy as np

temp=[14.3,14.1,14.1,14.2,14.1,13.7,13.8,14.1,14.2,14.1, 13.8,14.1,8,8,
      8.2,8.3,7.1,6.7,6.8,7,7.6,7.4,7.7,20.6,20.2,20.1,20.1,20.3,20.4,
      20.9,21.4,21.4,20.6,20.3,20.5]
listnp=np.array(temp)

print listnp
print np.mean(listnp)
print np.std(listnp)
```

Figura 28: Código para sacar la media y la std dev

A partir de todas las temperaturas de la Figura 29, sale una mediana de 14,2342857143 (14,23) y una desviación estándar de 5,29759745314 (5,29) así que crearemos nuestro random dataset a partir de esos valores.

	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiem- bre	Octubre	Noviem- bre	Diciem- bre
Temperatura media (°C)	14.3	14.1	14.1	14.2	14.1	13.7	13.8	14.1	14.2	14.1	13.8	14.1
Temperatura mín. (°C)	8	8	8.2	8.3	8	7.1	6.7	6.8	7	7.6	7.4	7.7
Temperatura máx. (°C)	20.6	20.2	20.1	20.1	20.3	20.4	20.9	21.4	21.4	20.6	20.3	20.5

Figura 29: Valores temperaturas en Villaflora. Fuente: www.climate_data.org

Como los valores de temperaturas son aleatorios, y los instantes están ordenados secuencialmente para facilitar el análisis (mucho más lógico), puede pasar que en un determinado instante la temperatura sea una, y seguidamente sea otra muy diferente (se asume este error ya que sólo se dará en el prototipo, no debe darse en ningún caso cuando se tengan datos reales).

Generación de datos

Cada vez que se importen datos del sensor de corriente, la columna de tiempo se actualizará, así que a su vez también se crearan datos de temperatura con la columna de tiempo actualizada.

A través de este simple comando desde la terminal se actualizará el archivo y permitirá trabajar con nuevos datos.

```
Generar data a través de: curl "https://api.mockaroo.com/api/edd0f0b0?count=1000&key=07755790" > "Temperatura ext.csv"
```

Temperatura interior

Definición de valores y umbrales

Temperatura int

Save Changes

Field Name	Type	Options
row	Row Number	blank: 0 % <input type="checkbox"/> <input type="checkbox"/>
last_val_time	Dataset Column	Current Sensor <input type="checkbox"/> time <input type="checkbox"/> sequential <input type="checkbox"/> blank: 0 % <input type="checkbox"/> <input type="checkbox"/>
ave	Normal Distribution	mean: 25 std. dev: 15 decimals: 2 blank: 0 % <input type="checkbox"/> <input type="checkbox"/>

Figura 30: Creación dataset temperatura interior

Hay que tener en cuenta que en un mismo nodo no se darán las temperaturas extremas. Es decir, un nodo situado en zona cálida es probable que llegue a temperaturas alrededor de los 50°C en horas de sol, pero nunca a temperaturas tan bajas como 0°C.

Por lo contrario, un nodo situado a gran altitud y en zona de clima frío, puede rozar temperaturas negativas en días fríos y con ausencia de sol.

Pero para poder jugar ahora con todos los valores, el valor de la temperatura interior se genera a partir de una distribución normal de datos con media de 25 y desviación estándar de 15.

Generación de datos

```
curl "https://api.mockaroo.com/api/b22af850?count=1000&key=07755790" > "Temperatura int.csv"
```

Humedad

Definición de valores y umbrales

Humedad

Save Changes

Field Name	Type	Options
row	Row Number	blank: 0 % <input type="checkbox"/> <input type="checkbox"/>
last_val_time	Dataset Column	Current Sensor <input type="checkbox"/> time <input type="checkbox"/> sequential <input type="checkbox"/> blank: 0 % <input type="checkbox"/> <input type="checkbox"/>
ave	Normal Distribution	mean: 40 std. dev: 10 decimals: 2 blank: 5 % <input type="checkbox"/> <input type="checkbox"/>

Figura 31: Creación dataset humedad

Como ya se mencionó los umbrales de trabajo deben estar entre el 30% y el 55%.

Los valores que muestra la columna *value* del dataset, provienen también de una distribución normal con media de 40% y desviación estándar de 10.

Generación de datos

```
curl "https://api.mockaroo.com/api/d93b7080?count=1000&key=07755790" > "Humedad.csv"
```


Energía eléctrica

Definición de valores y umbrales

Energía eléctrica

[Save Changes](#)

Field Name	Type	Options
row	Row Number	blank: 0 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
last_val_time	Dataset Column	Current Sensor <input checked="" type="checkbox"/> time <input checked="" type="checkbox"/> sequential <input type="checkbox"/> blank: 0 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
salida1	Normal Distribution	mean: 115 std. dev: 5 decimals: 2 blank: 8 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
salida2	Normal Distribution	mean: 114 std. dev: 4 decimals: 2 blank: 5 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
salida3	Normal Distribution	mean: 113 std. dev: 5 decimals: 2 blank: 6 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
salida4	Normal Distribution	mean: 118 std. dev: 6 decimals: 2 blank: 5 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
salida5	Normal Distribution	mean: 120 std. dev: 4 decimals: 2 blank: 3 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
salida6	Normal Distribution	mean: 116 std. dev: 5 decimals: 2 blank: 7 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
salida7	Normal Distribution	mean: 115 std. dev: 6 decimals: 2 blank: 5 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
salida8	Normal Distribution	mean: 119 std. dev: 3 decimals: 2 blank: 9 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>

Figura 32: Creación dataset energía eléctrica

Los valores del consumo eléctrico deberían ser bastante constantes en el tiempo. Cada salida puede dar entre 110-125 VAC (voltios de corriente alterna), así que teniendo en cuenta que hay 8 salidas en cada regleta de alimentación, hay que simular ocho columnas de valores dentro del rango especificado.

Es interesante tratar cada salida por separado ya que si una presentase algún problema de voltaje, sería rápidamente detectado.

Generación de datos

```
curl "https://api.mockaroo.com/api/7694a710?count=1000&key=07755790" > "Energía eléctrica.csv"
```

Movimiento

Definición de valores y umbrales

Motion&Door

[Save Changes](#)

Field Name	Type	Options
row	Row Number	blank: 0 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
last_val_time	Dataset Column	Current Sensor <input type="checkbox"/> time <input checked="" type="checkbox"/> sequential <input type="checkbox"/> blank: 0 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
sensor	Number	min: 1 max: 2 decimals: 0 blank: 0 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
event	Boolean	blank: 0 % <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>

Figura 33: Creación dataset movimiento

La columna de sensor tiene dos posibles valores diferentes:

- 1: refiriéndose al sensor1, que en éste caso será el que detecta movimiento dentro del nodo, Wall Mount Motion Sensor.
- 2: refiriéndose al sensor2, el encargado de detectar si hay apertura de puerta, el Door Sensor.

A partir de aquí, se crean valores booleanos aleatorios (*True*, *False*), que indicaran si en el momento de toma de medida (columna importada del dataset de los sensores de corriente) se ha detectado movimiento/apertura de puerta o no.

Generación de datos

```
curl "https://api.mockaroo.com/api/6e77b360?count=1000&key=07755790" > "Motion&Door.csv"
```

6.2 Algoritmo y análisis en Python

6.2.1 Python

Python es un lenguaje de programación de propósito general, una apuesta por la simplicidad, versatilidad y rapidez de desarrollo. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

En los últimos años el lenguaje se ha hecho muy popular, como se puede observar en la *Figura 34* gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que se puede desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.
- Además, Python es gratuito, incluso para propósitos empresariales.

A parte de todas las ventajas comentadas, Python es el lenguaje utilizado en las asignaturas de programación de la universidad, cosa que facilitó crear el código ya que se disponía de una buena base.

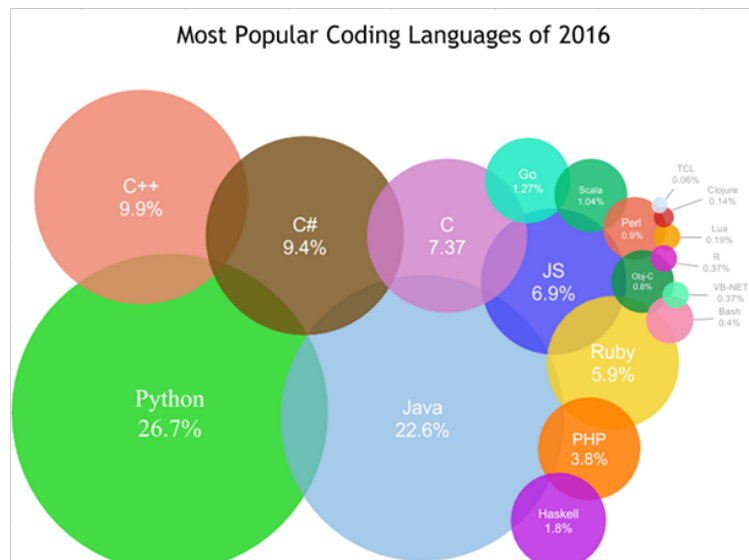


Figura 34: Gráfico popularidad de los lenguajes de programación 2016. Fuente: blog Codeeval

6.2.2 Herramientas y librerías

Como en todo lenguaje de programación, hay miles de librerías y herramientas que te permiten tratar con todo tipo de temas.

Obviamente hay ciertas librerías que son comunes y necesarias sea cual sea el tema que estas tratando, como serian datetime (para tratar con fechas), math, random...

En el caso concreto de este proyecto, cobran importancia las herramientas que permiten tratar con una gran cantidad de datos y hacen que el análisis sea mucho más simple y visual.

La más potentes y destacables entre todas son:

- NumPy
- Pandas

NumPy

NumPy es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.

Concretamente en este proyecto se utilizará la librería NumPy para tratar los DataFrames (que se verán a continuación), p.e. para imponer condiciones a las columnas, filas o números que cumplan ciertos requisitos.

También será imprescindible para llamar la librería **matplotlib.pyplot** [6]. Incluida dentro de NumPy, es la encargada de crear figuras y gráficos. Puedes escoger qué tipo de gráfico quieres, así como editarlo según las necesidades.

Pandas

Pandas es una librería destinada a la manipulación de datos que ofrece estructuras y operaciones para manipular tablas numéricas y series temporales.

Su principal herramienta es el objeto DataFrame que básicamente consiste de tres elementos: datos, índice y columnas. Cada fila de un DataFrame corresponde a medidas o valores de algo, mientras que cada columna es un vector que contiene data para una variable específica. Eso significa que cada fila no tiene, pero puede, contener el mismo tipo de valores: estos pueden ser numéricos, caracteres, lógicos, etc.

Así pues, se definen como estructuras de datos bidimensionales con columnas de diferentes tipos.

6.2.3 Algoritmo

Como ya se ha visto anteriormente, tanto los datos que provienen de la base de MongoDB (con los que se trabajará totalmente en un futuro), como los que provienen de los datasets provisionales, llegan en un formato no muy cómodo de visualizar.

Es por este motivo que era necesario crear unos archivos con un código en Python para poder tratar los datos, limpiarlos, destacar los parámetros importantes, clasificar las alarmas y ver posibles relaciones.

Cada algoritmo generado sirve para tratar un dataset distinto, pero hay algunas líneas de código que sirven para todos:

- Abrir el documento .csv
- Renombrar los campos estudiados para que sean más claros
- Conversión de tiempo EPOCH a HUMAN TIME
- Hay ciertos instantes de tiempo en que los sensores no envían información. Estos instantes se completan automáticamente con valores NaN. Para simplificar el estudio y aproximarlos a la realidad, se completará este campo con el mismo valor de un instante anterior (método *ffill* de pandas)
- Siempre es más interesante poder visualizar estos datos de forma gráfica, para ver su evolución en el tiempo y si siguen alguna distribución específica. A través de Matplotlib es posible ver estas gráficas.
En este apartado se incluyen comandos para adaptar el eje de las x a formato fecha, también se da nombre al gráfico, ejes y se indica cada cuanto tiempo se marcará un punto en el gráfico.
- Hay programadas ciertas variables que te dan información sobre ciertos parámetros importantes, así como la media y los valores máximo y mínimo de todas las medidas.
- Y como último elemento común, cuando se sobrepasan los umbrales definidos anteriormente, se almacena esta información en un DataFrame de alarmas que posteriormente se tratará junto con los otros archivos.

Temperatura exterior

En este archivo sólo hay los comandos básicos explicados anteriormente.

No se programó nada en referencia a las alarmas ya que no se puede actuar sobre la temperatura externa.

Temperatura interior

En este archivo se pueden diferenciar dos partes claras: la primera está diseñada, como en los otros casos, para cuando se tengan valores reales de los sensores. Pero hay otra parte que se ha añadido con el objetivo de aproximarse un poco más a la realidad si se trabaja con valores aleatorios.

Esta modificación tiene en cuenta los valores de temperatura exterior, haciendo que si la temperatura exterior es mayor a 17 grados centígrados, la temperatura interior pase a ser la siguiente:

$$\text{Temperatura interior} = \text{Temperatura interior} + \text{random}(0.5, 1.5)$$

Esta condición se basa en la suposición de que en horas de sol o cuando la temperatura exterior es alta (en este caso se considera alta cuando es superior a 17°C), la temperatura en el *shelter* aumenta y los equipos tienden a recalentarse más de lo normal. Se consigue este aumento de temperatura a través de la función *random*, que creará números aleatorios comprendidos entre el 0.5 y el 1.5 y le sumará este valor al de temperatura interior inicial.

Por lo contrario, cuando la temperatura exterior es inferior a este valor, se supondrá que es de noche o que el clima es frío, por lo que la temperatura en el *shelter* disminuirá y los equipos no se recalentarán. Así pues:

$$\text{Temperatura interior} = \text{Temperatura interior} - \text{random}(0.5, 1.5)$$

Esto se hace a partir del dataset creado de Temperatura interior, en el cuál se añade una columna con los valores del dataset de temperatura exterior, quedando así tres columnas: la del instante de medida, "value" que hace referencia a los valores de la temperatura interior y "Tempeext" con los valores de la temperatura exterior. A partir de aquí, se modifica la temperatura de "value" siguiendo las indicaciones dadas anteriormente.

Cabe comentar también que en el momento de graficar los valores, se añade un pequeño detalle para hacerlo más visual. Se marca con un círculo azul los valores que sobrepasan los 40°C, como así indica el límite del parámetro, comentados en el apartado *Límites del parámetro en 4.1.5 Temperatura interior*. También ocurrirá algo parecido, pero en este caso indicado con círculos rojos, cuando la temperatura esté por debajo de los 0°C.

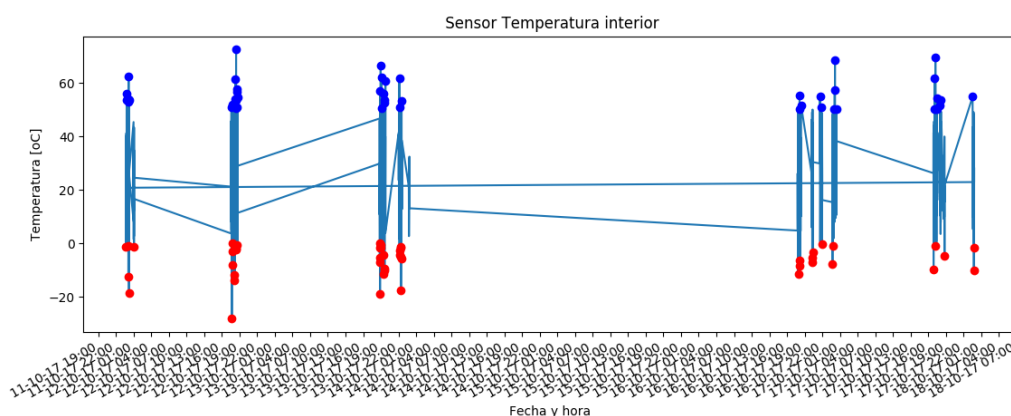


Figura 35: Gráfico valores temperatura interna

En este caso sí que hay que tener en cuenta las alarmas cuando se superan ciertos valores. Es por eso que, como se ve en la Figura 35, se crea un DataFrame (*alarmast*) que sigue los siguientes pasos:

Desde el DataFrame de la temperatura interior, se filtran los valores que superan los límites indicados, creando así nuevas columnas que indicaran, los instantes en qué ocurre este evento, valores, tipo de alarma (puede ser que la temperatura supere el límite superior o el inferior) y finalmente, se ordena todo temporalmente.

```
#CREAR ALARMA
#DataFrame de la alarma 1
alarmas1=df2[df2['value'] > 40]
alarmas1['tiposensor']= 'Temperatura'
alarmas1['event']= 'Temperatura H'
del alarmas1['Tempext']

#DataFrame de la alarma 2
alarmas2=df2[df2['value'] < 0]
alarmas2['tiposensor']= 'Temperatura'
alarmas2['event']= 'Temperatura L'
del alarmas2['Tempext']

#Juntamos DataFrames en uno solo
alarmast=alarmas1.append(alarmas2, ignore_index=True)

#Ordenamos las alarmas por tiempo
alarmast=alarmast.sort_values(by='time')
```

Figura 36: Código alarma temperatura interior

Humedad

A parte del trozo de código general, tanto los gráficos como las alarmas tienen la misma estructura que en el caso de la temperatura interna, siendo ‘Humedad H’ cuando el valor > 55% y ‘Humedad L’ cuando el valor sea < 30%.

Energía eléctrica

La dificultad añadida que presenta este apartado es que la alarma se puede disparar debido a cualquier de las ocho salidas que presenta la regleta de alimentación. Es por eso que será necesario detectar a que salida se debe el aviso y registrar su valor para poder almacenarlo correctamente en el DataFrame de alarmas (*alarmase3*).

Si en un mismo instante hay dos salidas que sobrepasan el límite (ya sea inferior o superior), se almacenarán como dos alarmas distintas.

Para todo lo mencionado anteriormente, el procedimiento seguido es crear dos DataFrames por separado, primero uno con todos los valores detectados que superan el límite superior y dándoles una etiqueta de “Voltaje H” y seguidamente, uno con todos los que son inferiores al límite inferior, llamados pues “Voltaje L”. Finalmente se fusionan en uno y quedan ordenados temporalmente.

También en el apartado dónde se detectan los valores importantes se añade una cosa. Primero de todo se crean tres listas, donde se van acumulando los valores máximo, mínimo y medio de cada salida. Finalmente, a través de una función que recorre estas listas, es posible saber cuál es valor máximo, mínimo y medio global.

Movimiento

Los sensores de movimiento trabajan de forma un poco distinta a los otros. Se filtrarán las filas cuyo valor en la columna de *event* sea *True*, ya que significa que el sensor ha detectado algo, ya sea movimiento o apertura de puerta.

También se diferencia en el método de como llenar los campos con valores NaN. En vez de fijarse en el valor del instante anterior, automáticamente los instantes que no presenten valor pasaran a ser *False*. Eso se debe a que la probabilidad de que ocurra dicho evento es menos probable que la probabilidad de que realmente se detecte algo.

Se puede observar en el código que no hay variables referentes a información destacable, pues se sustituyeron por una variable que nos indica el recuento de veces que ha ocurrido un evento.

En la parte de graficar, el gráfico escogido para mostrar que está ocurriendo en el nodo por lo que respecta al movimiento es un gráfico circular. Indica el % de veces que el sensor da el valor de *True*, sobre el total de señales (tanto *True* como *False*) que han llegado.

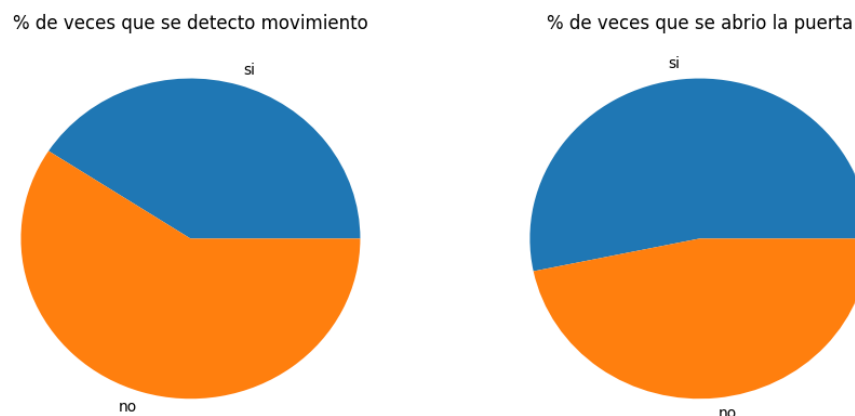


Figura 37: Gráfico que representa el sensor de movimiento

Sensor corriente

Al ser este el único sensor del que disponemos datos reales, su código es el que se aproxima más a lo que acabará siendo el modelo final para todos los anteriores.

Igualmente, los campos generales son los mismos y por lo que respecta a los gráficos, se sigue el mismo modelo utilizado en la temperatura interior y humedad, indicando los valores que están sobrepasando los límites, tanto del sensor 1 como del sensor 2. Se utiliza el color rojo para el sensor 1 y el azul para el sensor 2.

La diferencia con los otros códigos está en que en este caso no es necesario el tratamiento exhaustivo de las alarmas. Directamente de la base de MongoDB se extrae el documento que contiene las

alarmas, eliminando aquellas que no nos interesan (como serían las conexiones o desconexiones de los mPort, etc.), así nos queda ya un DataFrame con los mismos campos que los DataFrames de las otras alarmas.

Eventos y alarmas

Finalmente, a pesar de que en cada archivo se crea un DataFrame con las alarmas que se disparan de cada parámetro, es interesante disponer de un documento que lo resuma todo, lo ordene y permita buscar en él cualquier detalle al respecto.

Así que en este fichero básicamente lo que se hace es importar los DataFrames de las alarmas de cada parámetro (*alarmast*, *alarmash*, *alarmase3*, *alarmasm* y *alarmasc*) y juntarlos de nuevo en un DataFrame global (*TOT*), se ordenan temporalmente y se exporta a un fichero excel (*eventsexport.xlsx*). En este fichero están incluidas las del sensor de corriente, que se han adaptado al formato estándar de las otras para que al juntarlo no den problemas de formato.

Por otro lado, tenemos los eventos y alarmas reales que provienen directamente del mFi Controller. *Eventssensor.py* es el fichero que servirá cuando ya se tenga toda la plataforma instalada, así que aunque ahora mismo solo se obtengan los datos del sensor de corriente, el mismo código ya sirve para cuando esté todo listo. En este caso, se adquieren las columnas que nos aportan información interesante, se eliminan las filas que contienen NaN como valor y se hace otra vez la conversión EPOCH a HUMAN TIME. Finalmente, se exporta también a un fichero excel (*EventsSensorexport.xlsx*).

Desde el mFi Controller Software no hay que programar ninguna línea de código ya que el programa te lo permite hacer de forma sencilla. Los pasos a seguir son los siguientes:

- Indicar nombre de la alarma y a que sensor corresponde
- En qué valor debe saltar la alarma (y si hay que mantenerse un cierto tiempo antes de saltar)
- Qué acciones se deben tomar (mandar mail en este caso)
- Concretar cuanto tiempo se debe esperar hasta crear nuevo evento

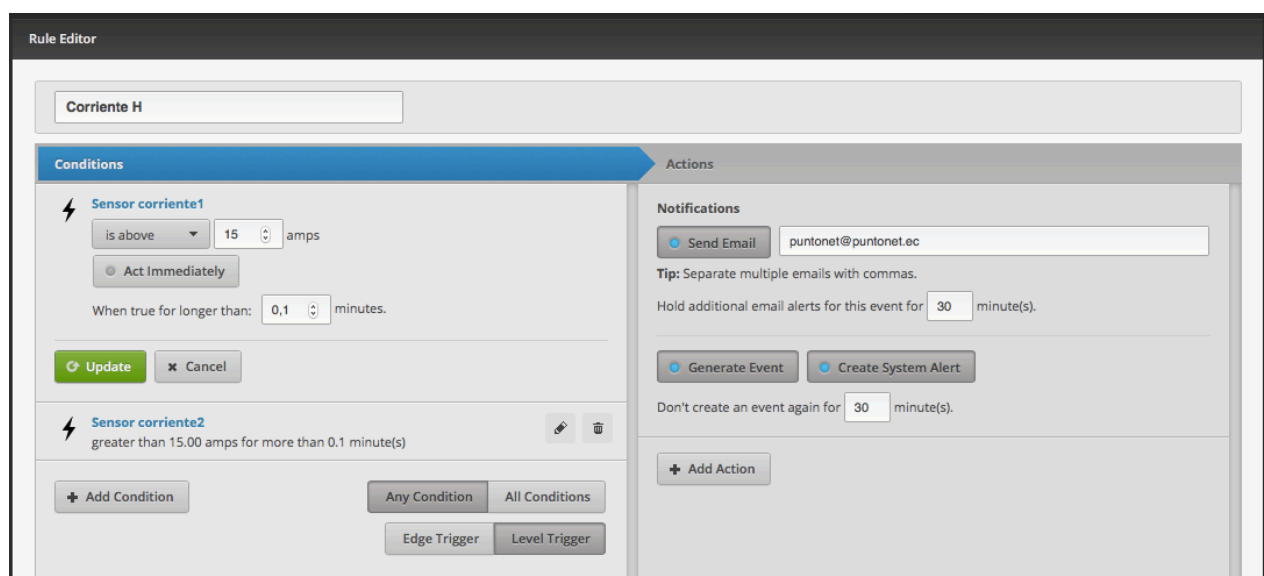


Figura 38: Interfaz Rules Tab con la alarma del sensor de corriente (límite superior)

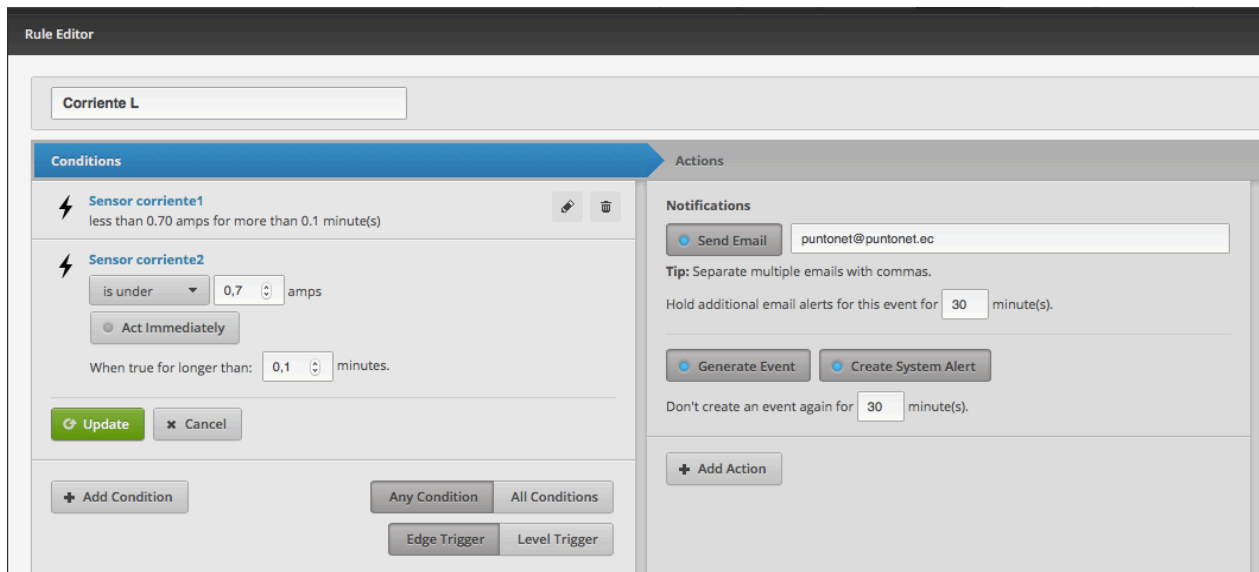


Figura 39: Interfaz Rules Tab con alarma del sensor de corriente (límite inferior)

Relaciones

Se plantean diferentes hipótesis sobre cuales podrían ser las posibles relaciones entre variables.

Se estudiarán a través del coeficiente de correlación lineal de Pearson (ρ), el cual está pensado para variables cuantitativas y mide el grado de covariación entre dos variables relacionadas linealmente (sin depender de la escala de cada variable, ya que el coeficiente es la covarianza dividido entre el producto de las desviaciones típicas).

$$\rho = \frac{\text{Covariancia}}{\prod \text{desviaciones típicas}}$$

Ecuación 2: Fórmula coeficiente correlación

Digamos, en primera instancia, que sus valores oscilan entre -1 y 1. No obstante ha de indicarse que la magnitud de la relación viene especificada por el valor numérico del coeficiente, reflejando el signo la dirección de tal valor. En este sentido, tan fuerte es una relación de +1 como de -1. En el primer caso la relación es perfecta positiva y en el segundo perfecta negativa. Se dice que la correlación entre dos variables X e Y es perfecta positiva cuando exactamente en la medida que aumenta una de ellas aumenta la otra y perfectamente negativa cuando una disminuye en la medida que disminuye la otra.

Aplicando esta teoría a las posibles relaciones que se podrían obtener, está programado en el código el ρ entre corriente y temperatura interna, ya que si aumenta el corriente que alimenta los equipos del nodo, estos pueden calentarse y aumentar de temperatura.

También, para asegurar que se cumple algo lógico, se calcula el ρ entre las dos temperaturas, la exterior y la anterior. Podría darse que no fuera una correlación perfecta positiva o negativa ya que no tienen por qué variar en la misma medida.

De ninguna de estas relaciones se pueden sacar conclusiones reales ya que los datos no son reales de momento, sólo un prototipo.

Cabe decir que con los datos reales, y como propuesta futura, se podrían detectar muchas más relaciones, ya sea gracias a una relación clara a simple vista o para estudiar puntos críticos dónde los valores son atípicos.

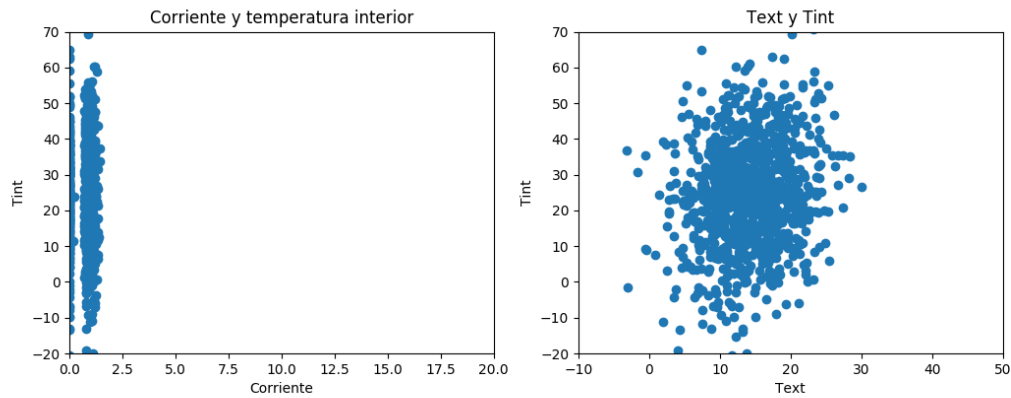


Figura 40: Gráfico scatter de las relaciones

7. PLANIFICACIÓN Y COSTOS

7.1 Planificación temporal

Durante el desarrollo del proyecto se ha ido valorando cuanto tiempo es necesario para desarrollar cada una de las tareas. A continuación, se puede observar el Diagrama de Gantt (*Figura 40 y Figura 41*), elaborado a través de la página web Tom's planner, y estructurado de la siguiente forma:

El eje horizontal muestra el tiempo dedicado a cada tarea en formato de días de cada mes que se ha trabajado en el proyecto (aproximadamente entre el 21 de agosto de 2017 hasta diciembre de 2017).

En el eje vertical hay un listado de las actividades que se han llevado a cabo, ordenadas en cinco grandes grupos. El primero corresponde a lo que sería la definición y formación previa al inicio del proyecto. Al ser un proyecto del ámbito de las telecomunicaciones, se puede considerar una etapa esencial para adquirir los conocimientos necesarios y básicos.

Pasando a la siguiente fase, hay la etapa de implementación, en la que se agrupan todas las actividades referentes al apartado de *software* y *hardware*, así como la primera fase de tratamiento de datos. Durante todo este tiempo y de forma paralela, se lleva a cabo un autoaprendizaje de las librerías de Python enfocadas a lo que sería el datamining mediante cursos online y/o proporcionados por la empresa.

Eso nos lleva al grupo de programación, que una vez se tienen unas primeras nociones sobre los datos con los que se va a trabajar, se pasan a crear los algoritmos y datasets necesarios que en un futuro servirían para sacar las conclusiones y para cumplir con los objetivos iniciales.

De forma paralela se llevan a cabo tareas como la instalación del sensor en el nodo y se estudian las posibilidades para dejar lista la conexión entre el nodo y la oficina. Esta tarea, como es lógico, no necesita de las del grupo de programación.

Y finalmente, la fase de redacción y preparación de la exposición oral donde está la base del proyecto: la redacción de la memoria. Esta fase es importante en cualquier proyecto pero cabe destacar que en este es sumamente importante. Esto se debe a que uno de los objetivos principales es que el diseño de toda la plataforma debe servir como manual para futuras instalaciones de la misma plataforma, es por eso que tenerlo todo bien redactado y estructurado es esencial.

Así pues, desde el inicio hasta el final del proyecto se han necesitado unos 3 meses y medio, teniendo en cuenta que hay días en los cuáles la carga de trabajo es superior a otros.

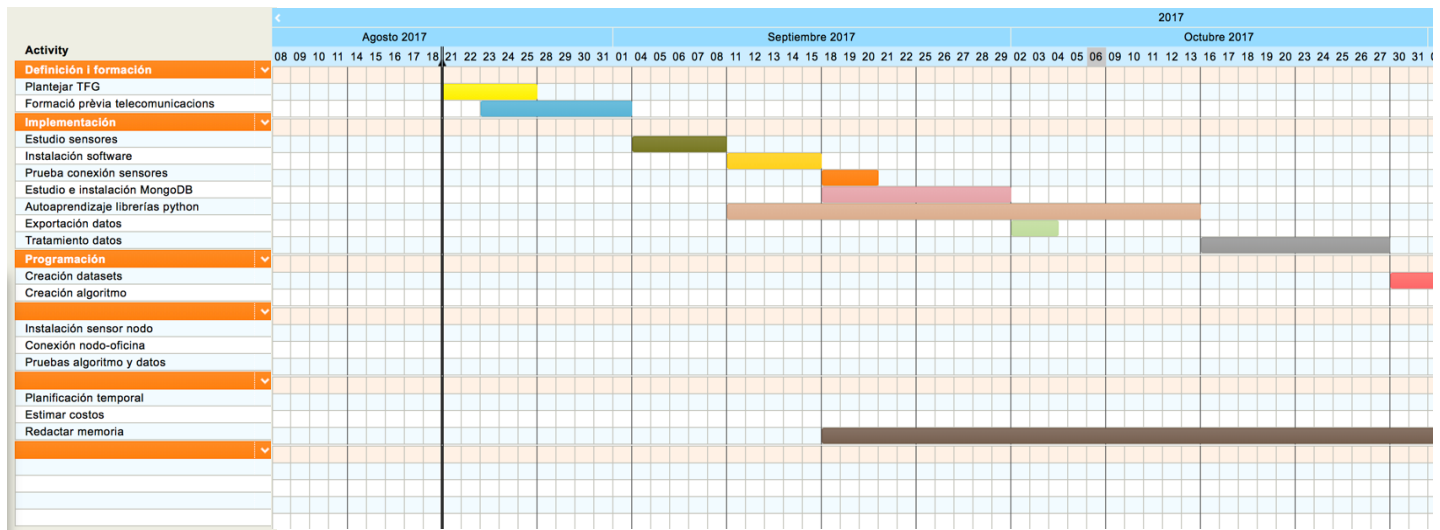


Figura 41: Diagrama de Gantt parte 1

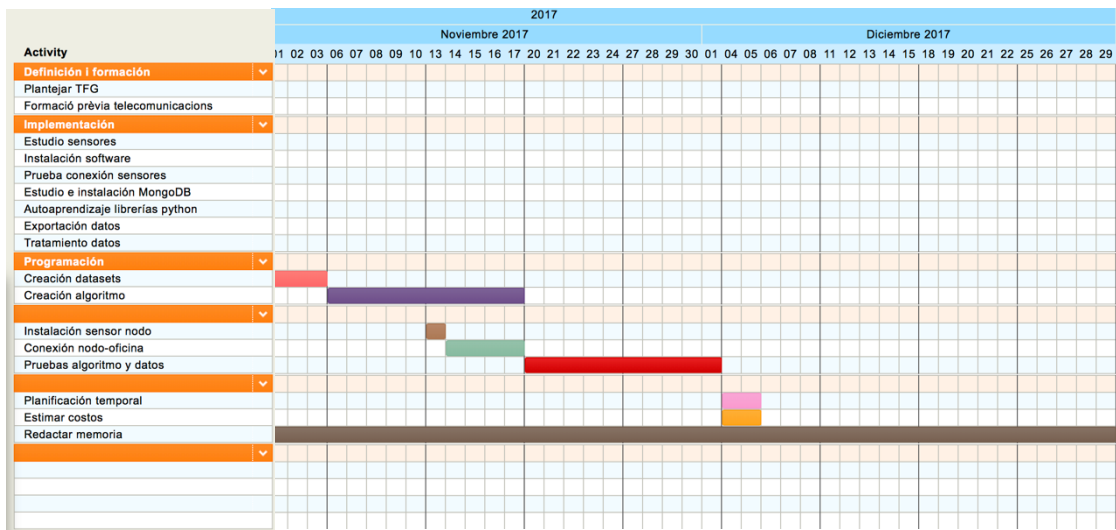


Figura 42: Diagrama de Gantt parte 2

7.2 Costos

Una vez se ha calculado el tiempo necesario para realizar el proyecto, se puede hacer una buena aproximación del costo general de este.

Empezando por lo que supondría el contrato de un ingeniero junior: el desarrollo del proyecto tuvo lugar durante unas prácticas en la empresa Puntonet, donde la jornada laboral era de 8h/día, pero es más aproximado decir que en el proyecto se trabajaron unas 5 horas de media, ya que no todas las horas de la jornada laboral se dedicaban a eso y aunque algún día sí, también hubo días incluidos en el anterior diagrama de Gantt que no se pudo avanzar nada.

Se calculan unos 20 días de trabajo por mes, 3 meses y medio en total y un sueldo de 25€/h para un ingeniero sin o con poca experiencia previa como sería en este caso. Quedaría:

$$3,5 \text{ meses} \times 20 \frac{\text{días}}{\text{mes}} \times 5 \frac{\text{h}}{\text{día}} \times \frac{25\text{€}}{\text{h}} = 8750 \text{ €}$$

Ecuación 3: Cálculo costo ingeniero junior

Cabe destacar que si se calculan las horas trabajadas, sale de unas 350h, cosa que cuadra perfectamente con las que se debería dedicar al TFG si lo calculas mediante los créditos que le corresponden y las horas estimadas por crédito.

$$12 \text{ créditos ECTS} \times \frac{28\text{h}}{\text{crédito}} = 336 \text{ h}$$

Ecuación 4: Cálculo horas a dedicar al TFG

Juntamos el costo referente a los dos tutores pendientes del trabajo, el de la universidad, que se debe tener en cuenta ya que el tiempo dedicado a supervisar y ayudar en el proyecto no es menospreciable, y el de la empresa, quién ayudó y llevo un seguimiento de todo el proceso.

Se asume que el sueldo de un ingeniero senior está en unos 40€/h en España y unos 35€/h en Ecuador:

$$\text{Tutor universidad: } 15\text{h} \times \frac{40\text{€}}{\text{h}} = 600 \text{ €}$$

$$\text{Tutor empresa: } 30\text{h} \times \frac{35\text{€}}{\text{h}} = 1\,050 \text{ €}$$

Ecuación 5 y 6: Cálculo costo tutores

En referencia al material utilizado contemplamos el ordenador, ya que es personal y las licencias de los programas utilizados. En el segundo caso, el costo sería de cero ya que todos los programas eran gratuitos o proporcionados por la empresa.

Precio MacBook Air: 1 350€

Se puede ver un resumen de todo lo mencionado en la tabla siguiente:

Elemento	Costo (€)
Ingeniero junior	8 750
Ingeniero senior España	600
Ingeniero senior Ecuador	1 050
Ordenador personal	1 350
Total	11 750

Tabla 7: Resumen costos proyecto

8. IMPACTO SOCIAL Y AMBIENTAL

8.1 Impacto social

La continua adaptación de las empresas en el ámbito de las nuevas tecnologías es muy importante y supone una mejora tanto a nivel económico como a nivel de identidad de empresa. Más importante es cuando esta empresa se dedica a las telecomunicaciones en un país como Ecuador, donde están tratando de seguir el ritmo mundial al que avanzan todos estos temas.

Es por eso que la diferencia entre las empresas de telecomunicaciones reside en temas como este. Está claro que actualmente los servicios de fibra óptica y sobretodo, de conexión Wifi están presentes en muchos países del mundo. Pero no todos son capaces de dar un buen servicio ni disponen de un sistema de monitorización que les ayude en esta tarea.

Así que destacaría el impacto social como un impacto a nivel de mentalidad de las empresas, una mentalidad que tiene que ser capaz de ver los resultados a largo plazo aunque estos supongan una gran inversión inicial. Porque es una sociedad que a pesar de disponer de las tecnologías más avanzadas, está acostumbrada a vivir el día a día y a invertir en negocios que den beneficios inmediatos. Es por eso que a veces cuesta arriesgarse para adoptar una nueva forma de trabajar si en el momento ya funciona bien lo que tienen.

La implementación de una plataforma de monitorización de los datos es un ejemplo de ello, un método que les permitirá solucionar problemas que ahora mismo no son vitales para sobrevivir como empresa pero que en un futuro, las ventajas serán incalculables.

8.2 Impacto ambiental

Un proyecto basado en la informática no tiene un elevado impacto ambiental. De normal, lo único que se podría destacar serían aspectos como el consumo energético del ordenador, el consumo de electricidad durante las horas de trabajo y por último, el transporte utilizado para llevar el sensor hasta la oficina, así como el transporte para llevar a cabo la tarea de instalación del sensor en el nodo.

Pero en este caso en concreto, se debe tener en cuenta que se trabaja con sensores, los cuales tienen un ciclo de vida determinado y hay que considerar las opciones de reciclaje.

Los sensores van conectados al mPort, el cuál está conectado al PoE y este a una fuente de alimentación, así que no hay que preocuparse por la duración de la batería. Pero como todo equipo electrónico no dura toda la vida. Actualmente, al ser unos equipos bastante nuevos, no hay referencias exactas sobre cuánto puede funcionar correctamente un equipo de la marca Ubiquiti pero sería importante tenerlo en cuenta. Además, el mPort es una pieza de 119 gramos de peso y está hecha básicamente de plástico así que una vez llegase a su fin, se debería seguir un proceso de reciclaje adecuado.

El sensor de corriente mFi-CS, un poco más pequeño que su compañero mPort, tiene un peso de 60 gramos y aparte de tener la estructura de plástico, contiene una pieza que actúa como bobina de alambre. Esta pieza acostumbra a ser de ferrita o hierro dulce, así que el proceso de reciclaje cambia.

CONCLUSIONES

En este proyecto finalmente se ha podido implementar la plataforma de monitoreo junto con el estudio de los datos.

Teniendo en cuenta los objetivos marcados en un inicio:

- La elección tanto de qué parámetros se iban a monitorear como de qué sensor que se iba a encargar de ello fue, en principio, correcta. Aunque eso no se puede asegurar al 100%, ya que no llegaron algunos sensores y en consecuencia, no se pudo evaluar de forma completa el funcionamiento de todos los sensores. Igualmente cabe destacar que los equipos que sí que llegaron y con los que se pudieron realizar pruebas, funcionaban como debían y como indicaba su respectivo manual de instrucciones.

La decisión de qué parámetros se iban a monitorear se ve afectada por el mismo problema. Al no tener datos reales, el estudio posterior de análisis de datos no dio resultados reales, por lo que no se vio si hacía falta algún parámetro más o si alguno de los escogidos resultaba no aportar nada.

- En referencia al *software* y al *hardware*, se podría decir que una vez se tiene toda la instalación preparada, el software funciona correctamente. A pesar de eso, sería interesante probar otros *softwares* más sencillos, ya que el momento de conectar los sensores al *software*, acostumbra a dar más problemas de los que debería. Sobre todo a nivel de conexiones y redes, ya que el proceso de instalación necesita un técnico experto en redes para solucionar ciertas etapas. Enfrentarse a esos problemas también ayudó mucho a ampliar los conocimientos más técnicos y ganar confianza a la hora de configurar equipos y routers.

Como parte positiva en esta etapa, se podría decir que con el *software* instalado y los sensores adoptados por el *software*, es un programa muy visual que te permite tener un buen control y una buena organización del sistema.

- Ligado a lo anterior, el mismo programa complica la extracción de datos. A pesar de eso, ha sido interesante investigar otras formas externas de solventar este problema, y se ha visto que son igual o más eficaces, y que sirven para una gama más amplia de programas que almacenan datos y no tienen su propia forma de exportación de datos. Interesante también las diferentes alternativas a la hora de escoger en que formato sería más cómodo tratar los datos *a posteriori*.

- Finalmente, y después de juntar todos los pasos anteriores, llega el momento de hacer algo con estos datos. Como primera decisión a tomar dentro de este apartado era imprescindible escoger bien el lenguaje de programación, que en este caso fue Python. La conclusión es que es un lenguaje totalmente adecuado para lo que se quería hacer de *datamining* y que te da muchísimas comodidades y opciones.

Otra consecuencia de la *no llegada* de los sensores fue tener que improvisar una alternativa para así poder dejar preparado un prototipo aunque no fuese con datos reales. La valoración sobre los medios utilizados para poder crear esos datos es muy positiva, ya que se obtuvieron

unos *datasets* adecuados para seguir con el estudio y dejar listo el código para un futuro análisis. Es verdad que eso impidió llegar a sacar conclusiones sobre como varían los parámetros, etc., pero cumplió con el objetivo de diseñar el estudio.

Como propuesta de cara a un futuro, diría que llevar a cabo este estudio con datos reales, sería muy interesante a la hora de solucionar realmente los problemas, e incluso llegar a prever futuros comportamientos de los parámetros.

AGRADECIMIENTOS

Primero de todo, agradecer a IAESTE, asociación de estudiantes de intercambio de prácticas que me permitió ir a hacer prácticas a Puntonet, la empresa en la que desarrollé este proyecto.

En Puntonet, agradecer a todo el equipo del que formé parte durante 3 meses, los cuáles estaban siempre pendientes de cualquier ayuda que pudiese necesitar y dispuestos a contribuir en mi formación. Especial mención al tutor que tenía allí, Wilson Freire.

Por otra parte, agradecer también la atención y contribución de mi tutor, Lluís Solano, por su seguimiento y consejos en momentos esenciales para el desarrollo del proyecto.

Y finalmente a mi familia, por el apoyo durante el tiempo que estaba aquí en Barcelona acabando el proyecto.

BIBLIOGRAFÍA

Referencias bibliográficas

- [1] PUNTONET, *Instructivo instalación de nodos*. Ecuador, Quito. 2012
[file:///Users/mariafont/Dropbox/TFG/INS%20NOC%2001%20002%20INSTRUCTIVO%20DE%20INSTALACION%20DE%20NODOS.html]*. *[URL, fecha de consulta]
- [2] HONEYWELL, *Outside Temperature Sensors*. Alemania
[https://products.ecc.emea.honeywell.com/spain/pdf/en0b0722-ge51r1215.pdf]*. *[URL, fecha de consulta]
- [3] INSTITUTO TECNOLÓGICA SUDAMERICANO, *Cuarto de Telecomunicaciones*. 2009
[https://es.slideshare.net/guesta4d883/cuarto-de-telecomunicaciones-1166154?next_slideshow=1]*. *[URL, fecha de consulta]
- [4] DITECOM, *Monitorización ambiental*. España.
[http://www.ditecom.com/datasheets/monitorizacion_ip/termometro-ip-HWg-STE.pdf]*. *[URL, fecha de consulta]
- [5] UBIQUITI NETWORKS, *mFi mPorts Interface and Sensors*
[https://www.ubnt.com/mfi/mport/]*. *[URL, fecha de consulta]
- [6] PYBONACCI, *Tutorial de matplotlib.pyplot*. 2012
[https://pybonacci.org/wp-content/uploads/2012/08/tutorial-de-matplotlib-pyplotv0-1-201208311.pdf]*. *[URL, fecha de consulta]

Bibliografía complementaria

- [1] CISCO, Cisco Networking Academy, *Introduction to IoT*
[https://www.netacad.com/es/courses/intro-iot/]*. *[URL, fecha de consulta]
- [2] PONTIFICIA UNIVERSIDAD JAVERIANA, *Prototipo sistema de monitoreo para la infraestructura eléctrica de un nodo remoto de telecomunicaciones para la empresa Telefónica Colombia*. Bogotá, D.C.
[www.javeriana.edu.co/biblos/tesis/ingenieria/tesis102.pdf]*. *[URL, fecha de consulta]
- [3] TECNOLOGÍA FÁCIL, *Qué es Ethernet?*. 2014
[https://tecnologia-facil.com/que-es/que-es-ethernet/]*. *[URL, fecha de consulta]
- [4] TECNOLOGÍA FÁCIL, *Qué es LAN?*. 2014

- [https://tecnologia-facil.com/que-es/que-es-lan/]*. *[URL, fecha de consulta]
- [5] UBIQUITI NETWORKS, *mPort Quick Start Guide*. 2015
[https://dl.ubnt.com/guides/mfi/mFi_mPort_QSG.pdf]*. *[URL, fecha de consulta]
- [6] 1&1 DIGITAL GUIDE, *Tutorial de MongoDB*
[https://www.1and1.es/digitalguide/paginas-web/desarrollo-web/tutorial-de-mongodb-primeros-pasos/]*. *[URL, fecha de consulta]
- [7] ROBO3T, *Robomongo*
[https://robomongo.org]*. *[URL, fecha de consulta]
- [8] MOKAROO, *Creación de datasets*
[https://www.mockaroo.com]*. *[URL, fecha de consulta]
- [9] DATACAMP, *Data Manipulation with Python*
[https://www.datacamp.com/home]*. *[URL, fecha de consulta]
- [10] -, *Coeficiente de correlación lineal de Pearson*
[https://personal.us.es/vararey/adatos2/correlacion.pdf]*. *[URL, fecha de consulta]
- [11] TOM'S PLANNER, *Software de planificación*
[https://www.tomsplanner.es]*. *[URL, fecha de consulta]

ÍNDICE ANEXO

A. Datasets	2
I. Dataset real sensor corriente	2
.....	2
II. Datasets creados a partir de Mockaroo	4
Temperatura exterior	4
Temperatura interior	5
Humedad	6
Energía eléctrica	7
Movimiento	8
B. Algoritmo Python	9
I. Temperatura exterior.....	9
II. Temperatura interior	10
III. Humedad	11
IV. Energía eléctrica.....	13
V. Movimiento	18
VI. Sensor corriente.....	19
VII. Eventos y alarmas	20
VIII. Relaciones.....	21
C. Ficheros .xlsx creados en eventos y alarmas.....	22
I. Fichero eventsexport.xlsx (a partir de datasets de Mockaroo).....	22
II. Fichero EventsSensorexport.xlsx (creado a partir del mFi Controller Software).....	24

A. Datasets

I. Dataset real sensor corriente

mFiminite5.csv

	A	B	C	D	E	F	G	H	I	J
1	time,size,sld,max,min,ave,last_val,last_val_time									
2	1507758120000,4,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758175000									
3	1507758180000,6,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758237000									
4	1507758240000,6,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758299000									
5	1507758300000,5,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758350000									
6	1507758360000,6,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758412000									
7	1507758420000,6,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758477000									
8	1507758480000,7,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758536000									
9	1507758540000,6,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758598000									
10	1507758600000,6,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758659000									
11	1507758660000,5,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758711000									
12	1507758720000,6,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758773000									
13	1507758780000,6,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758835000									
14	1507758840000,6,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758896000									
15	1507758900000,5,ObjectId(59de90453004f138372d638f),0,0,0,0,1507758951000									
16	1507758960000,6,ObjectId(59de90453004f138372d638f),0,0,0,0,1507759012000									
17	1507759020000,6,ObjectId(59de90453004f138372d638f),0,0,0,0,1507759074000									
18	1507759080000,6,ObjectId(59de90453004f138372d638f),0,0,0,0,1507759136000									
19	1507759140000,6,ObjectId(59de90453004f138372d638f),1.0114,0,0.1685666666666667,1.0114,1507759197000									
20	1507759200000,6,ObjectId(59de90453004f138372d638f),1.2509,1.2275,1.2407,1.2275,1507759259000									
21	1507759260000,5,ObjectId(59de90453004f138372d638f),1.2343,1.2077,1.2266400000000002,1.2077,1507759310000									
22	1507759320000,6,ObjectId(59de90453004f138372d638f),1.2251,1.1985,1.2106166666666665,1.2128,1507759372000									
23	1507759380000,6,ObjectId(59de90453004f138372d638f),1.2116,1.192,1.2051333333333332,1.2081,1507759434000									
24	1507759440000,6,ObjectId(59de90453004f138372d638f),1.2151,1.1904,1.2029666666666665,1.2114,1507759495000									
25	1507759500000,6,ObjectId(59de90453004f138372d638f),1.2449,1.1861,1.1983499999999998,1.1861,1507759557000									
26	1507759560000,6,ObjectId(59de90453004f138372d638f),1.1811,1.1615,1.17394,1.1615,1507759617000									
27	1507759560000,3,ObjectId(59de95e33004f138372d655d),0.0183,0,0.0061,0.0183,1507759617000									
28	1507759620000,6,ObjectId(59de90453004f138372d638f),1.1679,1.1333,1.1496666666666668,1.1333,1507759679000									
29	1507759620000,6,ObjectId(59de95e33004f138372d655d),1.155,1.123,1.1337,1.1272,1507759679000									
30	1507759680000,6,ObjectId(59de90453004f138372d638f),1.1491,1.1309,1.14116,1.1309,1507759730000									
31	1507759680000,5,ObjectId(59de95e33004f138372d655d),1.136,1.1162,1.12772,1.1253,1507759730000									
32	1507759740000,6,ObjectId(59de90453004f138372d638f),1.1366,1.1175,1.1286833333333333,1.1175,1507759791000									
33	1507759740000,6,ObjectId(59de95e33004f138372d655d),1.124,1.1049,1.11295,1.1049,1507759791000									
34	1507759800000,6,ObjectId(59de90453004f138372d638f),1.1362,1.1088,1.1258166666666667,1.1211,1507759853000									
35	1507759800000,6,ObjectId(59de95e33004f138372d655d),1.1215,1.0965,1.1095,1.1049,1507759853000									
36	1507759860000,6,ObjectId(59de90453004f138372d638f),1.1339,1.0925,1.1113166666666667,1.1339,1507759915000									
37	1507759860000,6,ObjectId(59de95e33004f138372d655d),1.1178,1.0838,1.0967999999999998,1.1178,1507759915000									
38	1507759920000,6,ObjectId(59de90453004f138372d638f),1.1214,1.0914,1.1054833333333334,1.0914,1507759977000									
39	1507759920000,6,ObjectId(59de95e33004f138372d655d),1.1007,1.0728,1.0913166666666667,1.0846,1507759977000									
40	1507759980000,6,ObjectId(59de90453004f138372d638f),1.1069,1.0824,1.0964666666666665,1.0824,1507760039000									
41	1507759980000,6,ObjectId(59de95e33004f138372d655d),1.0917,1.0685,1.0805166666666668,1.0815,1507760039000									
...										
741	1508276460000,6,ObjectId(59de95e33004f138372d655d),1.0663,1.0351,1.0497500000000002,1.0351,1508276514000									
742	1508276520000,6,ObjectId(59de90453004f138372d638f),1.0613,1.0322,1.04645,1.0322,1508276576000									
743	1508276520000,6,ObjectId(59de95e33004f138372d655d),1.0515,1.0363,1.0429833333333332,1.038,1508276576000									
744	1508276580000,6,ObjectId(59de90453004f138372d638f),1.0664,1.0387,1.0497666666666667,1.0436,1508276638000									
745	1508276580000,6,ObjectId(59de95e33004f138372d655d),1.0582,1.0357,1.0458166666666666,1.0461,1508276638000									
746	1508276640000,5,ObjectId(59de90453004f138372d638f),1.0801,1.047,1.0589600000000001,1.0629,1508276690000									
747	1508276640000,5,ObjectId(59de95e33004f138372d655d),1.0706,1.0422,1.05364,1.0554,1508276690000									
748	1508276700000,6,ObjectId(59de90453004f138372d638f),1.0642,1.0389,1.0515999999999999,1.0642,1508276751000									
749	1508276700000,6,ObjectId(59de95e33004f138372d655d),1.0597,1.0301,1.0514333333333334,1.0592,1508276751000									
750	1508276760000,6,ObjectId(59de90453004f138372d638f),1.1179,1.0242,1.0613333333333332,1.046,1508276813000									
751	1508276760000,6,ObjectId(59de95e33004f138372d655d),1.1205,1.0277,1.0573166666666667,1.0277,1508276813000									
752	1508276820000,6,ObjectId(59de90453004f138372d638f),1.0633,1.035,1.0481833333333332,1.0633,1508276875000									
753	1508276820000,6,ObjectId(59de95e33004f138372d655d),1.0716,1.0223,1.04585,1.0716,1508276875000									
754	1508276880000,6,ObjectId(59de90453004f138372d638f),1.0535,1.0321,1.0417666666666667,1.0409,1508276937000									
755	1508276880000,6,ObjectId(59de95e33004f138372d655d),1.0468,1.0319,1.0415833333333333,1.0388,1508276937000									

mFievent.csv

	A	B	C	D
1	time,key,rulename,sensor,sensor_value			
2	1507755976462,EVT_AP_DiscoveredPending,,,			
3	1507756149178,EVT_AP_Adopted,,,			
4	1507757221460,EVT_AP_Connected,,,			
5	1507760264413,EVT_AP_Lost_Contact,,,			
6	1507762847262,EVT_AP_Connected,,,			
7	1507766155366,EVT_AP_Lost_Contact,,,			
8	1507822484463,EVT_AP_Connected,,,			
9	1507822202468,EVT_AP_RestartedUnknown,,,			
10	1507822493469,EVT_AP_Connected,,,			
11	1507823560951,EVT_AP_Lost_Contact,,,			
12	1507824165957,EVT_AP_Connected,,,			
13	1507824461044,EVT_AP_Lost_Contact,,,			
14	1507824498239,EVT_AP_Connected,,,			
15	1507826800523,EVT_AP_Lost_Contact,,,			
16	1507913304356,EVT_AP_Connected,,,			
17	1507913148363,EVT_AP_RestartedUnknown,,,			
18	1507913312195,EVT_AP_Connected,,,			
19	1507914718200,EVT_AP_Lost_Contact,,,			
20	1507915328069,EVT_AP_Connected,,,			
21	1507916997061,EVT_AP_Lost_Contact,,,			
22	1507924942367,EVT_AP_Connected,,,			
23	1507927098586,EVT_AP_Lost_Contact,,,			
24	1507930265684,EVT_AP_Connected,,,			
25	1507930399358,EVT_AP_Lost_Contact,,,			
26	1507931122478,EVT_AP_Connected,,,			
27	1507931719449,EVT_AP_Lost_Contact,,,			
28	1508168889432,EVT_AP_Connected,,,			
29	1508168736439,EVT_AP_RestartedUnknown,,,			
30	1508168897471,EVT_AP_Connected,,,			
31	1508171276208,EVT_AP_Lost_Contact,,,			
32	1508177404450,EVT_AP_Connected,,,			
33	1508184158611,EVT_AP_Lost_Contact,,,			
34	1508190225805,EVT_AP_Connected,,,			
35	1508192859242,EVT_AP_Lost_Contact,,,			
36	1508251924257,EVT_AP_Connected,,,			
37	1508253509580,EVT_EMAIL_SendMailFailed,,,			

II. Datasets creados a partir de Mockaroo

Se muestran sólo los 40 primeros datos de los 1000 creados

Temperatura exterior

Nombre archivo: **Temperatura ext.csv**

	A	B
1	row,last_val_time,ave	
2	1,1507758175000,22.09	
3	2,1507758237000,11.42	
4	3,1507758299000,14.94	
5	4,1507758350000,12.86	
6	5,1507758412000,11.45	
7	6,1507758477000,17.5	
8	7,1507758536000,14.9	
9	8,1507758598000,12.68	
10	9,1507758659000,11.86	
11	10,1507758711000,20.13	
12	11,1507758773000,24.2	
13	12,1507758835000,12.74	
14	13,1507758896000,16.16	
15	14,1507758951000,19.89	
16	15,1507759012000,10.97	
17	16,1507759074000,13.84	
18	17,1507759136000,13.57	
19	18,1507759197000,17.24	
20	19,1507759259000,12.4	
21	20,1507759310000,10.84	
22	21,1507759372000,12.74	
23	22,1507759434000,13.3	
24	23,1507759495000,17.39	
25	24,1507759557000,15.38	
26	25,1507759617000,22.34	
27	26,1507759617000,21.52	
28	27,1507759679000,9.51	
29	28,1507759679000,5.58	
30	29,1507759730000,9.76	
31	30,1507759730000,14.26	
32	31,1507759791000,20.81	
33	32,1507759791000,23.61	
34	33,1507759853000,25.85	
35	34,1507759853000,20.41	
36	35,1507759915000,13.59	
37	36,1507759915000,15.3	
38	37,1507759977000,15.32	
39	38,1507759977000,13.7	
40	39,1507760039000,13.79	

Temperatura interior

Nombre archivo: **Temperatura int.csv**

	A	B
1	row,last_val_time,ave	
2	1,1507758120000,35.6	
3	2,1507758180000,32.31	
4	3,1507758240000,-1.31	
5	4,1507758300000,25.33	
6	5,1507758360000,41.19	
7	6,1507758420000,18.82	
8	7,1507758480000,13.12	
9	8,1507758540000,53.53	
10	9,1507758600000,34.65	
11	10,1507758660000,38.11	
12	11,1507758720000,31.76	
13	12,1507758780000,31.1	
14	13,1507758840000,9.64	
15	14,1507758900000,16.99	
16	15,1507758960000,9.71	
17	16,1507759020000,40.05	
18	17,1507759080000,21.86	
19	18,1507759140000,10.42	
20	19,1507759200000,23.19	
21	20,1507759260000,25.58	
22	21,1507759320000,8.32	
23	22,1507759380000,28.28	
24	23,1507759440000,23.09	
25	24,1507759500000,37.07	
26	25,1507759560000,-0.9	
27	26,1507759620000,16.24	
28	27,1507759680000,17.28	
29	28,1507759740000,27.02	
30	29,1507759800000,47.52	
31	30,1507759860000,36.22	
32	31,1507759920000,19.54	
33	32,1507759980000,28.16	
34	33,1507760040000,34.27	
35	34,1507760100000,21.47	
36	35,1507760160000,12.33	
37	36,1507760220000,39.74	
38	37,1507760280000,28.86	
39	38,1507760340000,16.31	
40	39,1507760400000,-18.64	

Humedad

Nombre archivo: **Humedad.csv**

	A	B
1	row,last_val_time,ave	
2	1,1507758120000,47.61	
3	2,1507758180000,44.34	
4	3,1507758240000,57.48	
5	4,1507758300000,51.89	
6	5,1507758360000,34.26	
7	6,1507758420000,26.46	
8	7,1507758480000,39.04	
9	8,1507758540000,	
10	9,1507758600000,39.21	
11	10,1507758660000,52.78	
12	11,1507758720000,40.37	
13	12,1507758780000,43.75	
14	13,1507758840000,43.84	
15	14,1507758900000,47.36	
16	15,1507758960000,43.06	
17	16,1507759020000,46.29	
18	17,1507759080000,50.82	
19	18,1507759140000,	
20	19,1507759200000,47.53	
21	20,1507759260000,38.5	
22	21,1507759320000,42.0	
23	22,1507759380000,45.45	
24	23,1507759440000,36.48	
25	24,1507759500000,36.78	
26	25,1507759560000,40.66	
27	26,1507759620000,47.11	
28	27,1507759680000,47.88	
29	28,1507759740000,51.02	
30	29,1507759800000,46.98	
31	30,1507759860000,34.75	
32	31,1507759920000,52.2	
33	32,1507759980000,47.91	
34	33,1507760040000,45.61	
35	34,1507760100000,38.41	
36	35,1507760160000,25.64	
37	36,1507760220000,43.52	
38	37,1507760280000,40.77	
39	38,1507760340000,44.15	
40	39,1507760400000,32.99	

Energía eléctrica

Nombre archivo: **Energia eléctrica.csv**

	A	B	C	D	E	F
1	row,last_val	time,salida1,salida2,salida3,salida4,salida5,salida6,salida7,salida8				
2	1,1507758120000	112.83,113.73,116.29,129.63,123.88,,119.32,116.79				
3	2,1507758180000	117.43,120.25,109.67,100.64,116.9,118.6,118.8,118.1				
4	3,1507758240000	113.93,111.43,108.47,117.21,122.83,114.7,112.9,115.97				
5	4,1507758300000	113.31,116.1,109.3,119.42,125.01,107.58,112.98,107.73				
6	5,1507758360000	114.85,115.17,111.21,120.46,119.3,122.92,123.86,				
7	6,1507758420000	118.31,108.7,112.28,110.62,116.48,120.31,119.31,117.34				
8	7,1507758480000	119.14,108.71,108.34,113.57,125.31,112.57,117.18,111.61				
9	8,1507758540000	105.67,115.67,115.86,127.17,117.6,119.04,121.55,123.29				
10	9,1507758600000	103.37,118.49,108.14,114.13,114.71,117.85,123.82,112.31				
11	10,1507758660000	121.77,116.9,113.2,115.89,114.81,117.69,120.58,120.35				
12	11,1507758720000	114.51,106.07,106.68,114.84,115.62,121.78,109.59,119.95				
13	12,1507758780000	120.08,117.31,115.63,118.61,123.38,121.27,127.03,				
14	13,1507758840000	115.39,111.56,118.36,115.43,112.83,107.92,				
15	14,1507758900000	123.96,111.74,111.26,125.08,118.15,117.93,122.05,113.31				
16	15,1507758960000	114.78,117.55,108.26,113.76,114.78,115.02,119.48,116.86				
17	16,1507759020000	109.98,111.9,120.17,115.66,110.78,111.85,124.64,118.22				
18	17,1507759080000	111.4,119.2,106.49,121.49,115.12,114.26,114.85,118.24				
19	18,1507759140000	111.99,115.22,112.25,115.89,121.11,104.68,107.5,118.07				
20	19,1507759200000	121.56,122.52,116.43,119.95,113.53,116.67				
21	20,1507759260000	116.65,107.46,101.11,121.77,117.5,114.16,117.36,116.77				
22	21,1507759320000	114.99,,108.73,,118.75,121.2,,119.33				
23	22,1507759380000	115.34,118.19,105.71,115.21,121.01,,114.95,119.65				
24	23,1507759440000	114.53,112.03,102.23,117.45,125.65,,114.5,116.16				
25	24,1507759500000	108.0,114.61,109.07,114.43,120.45,117.19,119.8,117.77				
26	25,1507759560000	115.38,110.47,108.2,120.58,120.19,112.67,110.58,113.49				
27	26,1507759620000	118.94,116.71,119.99,119.49,119.67,120.04,107.49,120.51				
28	27,1507759680000	119.21,118.8,110.17,114.21,122.65,123.92,113.89,119.34				
29	28,1507759740000	117.17,110.11,107.4,116.81,125.59,108.43,115.77,118.3				
30	29,1507759800000	116.67,117.81,109.86,129.68,118.72,115.75,125.31,118.18				
31	30,1507759860000	109.12,112.67,116.31,106.81,120.36,112.85,119.69,120.34				
32	31,1507759920000	115.01,,118.62,,127.92,112.63,,117.5				
33	32,1507759980000	108.79,116.16,115.2,113.47,121.36,115.4,110.14,116.38				
34	33,1507759940000	113.54,114.36,121.61,114.34,120.04,120.19,106.51,118.61				
35	34,1507759900000	109.86,115.26,104.79,121.97,124.0,,131.04,				
36	35,1507759860000	118.64,105.91,124.89,119.54,105.83,127.46,				
37	36,1507759820000	118.89,,117.09,,112.23,118.82,,116.79				
38	37,1507759780000	125.27,114.49,116.75,116.47,120.68,110.36,109.85,126.35				
39	38,1507759740000	124.7,115.76,,108.24,118.25,120.14,102.25,114.89				
40	39,1507759700000	118.17,113.68,111.36,119.16,116.92,119.45,113.81,112.86				

Movimiento

Nombre archivo: **Motion&Door.csv**

	A	B	C
1	row,last_val_time,sensor,event		
2	1,1507758120000,2,false		
3	2,1507758180000,1,false		
4	3,1507758240000,1,false		
5	4,1507758300000,1,false		
6	5,1507758360000,1,false		
7	6,1507758420000,2,true		
8	7,1507758480000,2,false		
9	8,1507758540000,2,false		
10	9,1507758600000,1,true		
11	10,1507758660000,1,false		
12	11,1507758720000,1,false		
13	12,1507758780000,1,false		
14	13,1507758840000,2,true		
15	14,1507758900000,2,true		
16	15,1507758960000,2,true		
17	16,1507759020000,1,false		
18	17,1507759080000,1,true		
19	18,1507759140000,2,true		
20	19,1507759200000,2,true		
21	20,1507759260000,2,true		
22	21,1507759320000,1,false		
23	22,1507759380000,1,false		
24	23,1507759440000,2,true		
25	24,1507759500000,1,true		
26	25,1507759560000,2,true		
27	26,1507759620000,2,false		
28	27,1507759680000,2,true		
29	28,1507759740000,1,false		
30	29,1507759800000,1,false		
31	30,1507759860000,2,true		
32	31,1507759920000,1,true		
33	32,1507759980000,2,false		
34	33,1507760040000,2,true		
35	34,1507760100000,1,false		
36	35,1507760160000,2,true		
37	36,1507760220000,1,false		
38	37,1507760280000,1,false		
39	38,1507760340000,2,false		
40	39,1507760400000,2,false		

B. Algoritmo Python

I. Temperatura exterior (tempext.py)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import time
import datetime
import matplotlib.dates as mdates

doc=pd.read_csv('Temperatura ext.csv')
df=pd.DataFrame(doc,columns=['row','last_val_time','ave'])
df.rename(columns={'last_val_time':'time','ave':'value'}, inplace=True)

#EPOCH TO HUMAN
df['time']=df['time']/1000
df['time'] = df['time'].apply(lambda x: datetime.datetime.fromtimestamp(x))

#COMPLETAR CAMPOS VACIOS (con el valor anterior)
df['value']=df['value'].fillna(method='ffill')

#GRAFICAR SENSORES
fig, ax = plt.subplots()
fig.autofmt_xdate()           #Para adaptar indice x a formato fecha

plt.xlabel('Fecha y hora')
plt.ylabel('Temperatura [oC]')
plt.title('Sensor Temperatura exterior')
plt.plot(df['time'], df['value'])

xfmt = mdates.DateFormatter('%d-%m-%y %H:%M')
ax.xaxis.set_major_locator(mdates.HourLocator(interval=3)) #to get a tick every 3 hours
ax.xaxis.set_major_formatter(xfmt)

#VALORES IMPORTANTES
maximo=df['value'].max()
minimo=df['value'].min()
media=df['value'].mean()

plt.close()
```

II. Temperatura interior (tempint.py)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import time
import datetime
import matplotlib.dates as mdates
import random
import math

doc1=pd.read_csv('Temperatura int.csv')
df1=pd.DataFrame(doc1,columns=['row','last_val_time','ave'])
df1.rename(columns={'last_val_time':'time','ave':'value'}, inplace=True)

#EPOCH TO HUMAN
df1['time']=df1['time']/1000
df1['time'] = df1['time'].apply(lambda x: datetime.datetime.fromtimestamp(x))

#COMPLETAR CAMPOS VACIOS (con el valor anterior)
df1['value']=df1['value'].fillna(method='ffill')

#GRAFICAR SENSORES
fig, ax = plt.subplots()
fig.autofmt_xdate() #Para adaptar indice x a formato fecha

plt.xlabel('Fecha y hora')
plt.ylabel('Temperatura [oC]')
plt.title(' Sensor Temperatura interior')
plt.plot(df1['time'], df1['value'])

xfmt = mdates.DateFormatter('%d-%m-%y %H:%M')
ax.xaxis.set_major_locator(mdates.HourLocator(interval=3)) #to get a tick every 3 hours
ax.xaxis.set_major_formatter(xfmt)
plt.plot(df1['time'][df1['value']> 50], df1['value'][df1['value'] > 50], 'bo') #Marcar con un
#circulo (bo= blue o) los puntos por encima de 40

plt.plot(df1['time'][df1['value']< 0], df1['value'][df1['value'] < 0], 'ro')

#VALORES IMPORTANTES
maximo=df1['value'].max()
minimo=df1['value'].min()
media=df1['value'].mean()
```

#AJUSTAR A VALORES REALES (eliminar cuando se tengan los sensores)

```
doc=pd.read_csv('Temperatura ext.csv')
df=pd.DataFrame(doc,columns=['row','last_val_time','ave'])
df.rename(columns={'last_val_time':'time','ave':'value'}, inplace=True)
df['value']=df['value'].fillna(method='ffill')
```

#ANADIMOS COLUMNA VALORES TEMP EXT

```
tempext=df['value']
df2=df1
df2['Tempext']=tempext
```

#Condicion de sumar o restar grados si la tempext es superior o inferior a 17 (condicion, valor si se cumple, valor si no se cumple)

```
df2['value'] = np.where(df2['Tempext']>17, df2['value']+ random.uniform(0.5,1.5),
df2['value']- random.uniform(0.5,1.5))
```

#CREAR ALARMA**#DataFrame de la alarma 1**

```
alarmas1=df2[df2['value'] > 40]
alarmas1['tiposensor']= 'Temperatura'
alarmas1['event']= 'Temperatura H'
del alarmas1['Tempext']
```

#DataFrame de la alarma 2

```
alarmas2=df2[df2['value'] < 0]
alarmas2['tiposensor']= 'Temperatura'
alarmas2['event']= 'Temperatura L'
del alarmas2['Tempext']
```

#Juntamos DataFrames en uno solo

```
alarmast=alarmas1.append(alarmas2, ignore_index=True)
```

#Ordenamos las alarmas por tiempo

```
alarmast=alarmast.sort_values(by='time')
```

```
plt.close()
```

III. Humedad (humedad.py)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import time
import datetime
```

```

import matplotlib.dates as mdates
import math

doc2=pd.read_csv('Humedad.csv')
df3=pd.DataFrame(doc2,columns=['row','last_val_time','ave'])
df3.rename(columns={'last_val_time':'time','ave':'value'}, inplace=True)

#EPOCH TO HUMAN
df3['time']=df3['time']/1000
df3['time'] = df3['time'].apply(lambda x: datetime.datetime.fromtimestamp(x))

#COMPLETAR CAMPOS VACIOS (con el valor anterior)
df3['value']=df3['value'].fillna(method='ffill')

#GRAFICAR SENSORES
fig, ax = plt.subplots()
fig.autofmt_xdate() #Para adaptar indice x a formato fecha

plt.xlabel('Fecha y hora')
plt.ylabel('Humedad Relativa (%)')
plt.title('Sensor Humedad')
plt.plot(df3['time'], df3['value'])

xfmt = mdates.DateFormatter('%d-%m-%y %H:%M')
ax.xaxis.set_major_locator(mdates.HourLocator(interval=3)) #to get a tick every 3 hours
ax.xaxis.set_major_formatter(xfmt)
plt.plot(df3['time'][df3['value']> 55], df3['value'][df3['value'] > 55], 'bo') #Marcar con un
circulo (bo) los puntos por encima de 40

plt.plot(df3['time'][df3['value']< 30], df3['value'][df3['value'] <30], 'ro') #Marcar con un
circulo (bo) los puntos por encima de 40

#VALORES IMPORTANTES
maximo=df3['value'].max()
minimo=df3['value'].min()
media=df3['value'].mean()

#CREAR ALARMA
#DataFrame de la alarma 1
alarmas1=df3[df3['value'] > 55]
alarmas1['tiposensor']= 'Humedad'
alarmas1['event']= 'Humedad H'

#DataFrame de la alarma 2
alarmas2=df3[df3['value'] < 30]

```

```
alarmas2['tiposensor']= 'Humedad'
alarmas2['event']= 'Humedad L'
```

#Juntamos DataFrames en uno solo

```
alarmash=alarmas1.append(alarmas2, ignore_index=True)
```

#Ordenamos las alarmas por tiempo

```
alarmash=alarmash.sort_values(by='time')
```

```
plt.close()
```

IV. Energía eléctrica (EnergiaElectrica.py)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import time
import datetime
import matplotlib.dates as mdates
import math
```

```
doc3=pd.read_csv('Energia Electrica.csv')
df4=pd.DataFrame(doc3,columns=['row','last_val_time','salida1','salida2','salida3', 'salida4',
'salida5', 'salida6','salida7', 'salida8'])
df4.rename(columns={'last_val_time':'time','ave':'value'}, inplace=True)
```

#EPOCH TO HUMAN

```
df4['time']=df4['time']/1000
df4['time'] = df4['time'].apply(lambda x: datetime.datetime.fromtimestamp(x))
```

#COMPLETAR CAMPOS VACIOS (con el valor anterior)

```
df4['salida1']=df4['salida1'].fillna(method='ffill')
df4['salida2']=df4['salida2'].fillna(method='ffill')
df4['salida3']=df4['salida3'].fillna(method='ffill')
df4['salida4']=df4['salida4'].fillna(method='ffill')
df4['salida5']=df4['salida5'].fillna(method='ffill')
df4['salida5']=df4['salida5'].fillna(method='ffill')
df4['salida6']=df4['salida6'].fillna(method='ffill')
df4['salida7']=df4['salida7'].fillna(method='ffill')
df4['salida8']=df4['salida8'].fillna(method='ffill')
```

#GRAFICAR SENSORES

```
fig, ax = plt.subplots()
fig.autofmt_xdate() #Para adaptar indice x a formato fecha
```



```
plt.xlabel('Fecha y hora')
plt.ylabel('Volatge VAC')
plt.title('Sensor mPower')
plt.plot(df4['time'], df4['salida1'], color='red')
plt.plot(df4['time'], df4['salida2'], color='blue')
plt.plot(df4['time'], df4['salida3'], color='green')
plt.plot(df4['time'], df4['salida4'], color='yellow')
plt.plot(df4['time'], df4['salida5'], color='orange')
plt.plot(df4['time'], df4['salida6'], color='pink')
plt.plot(df4['time'], df4['salida7'], color='black')
plt.plot(df4['time'], df4['salida8'], color='brown')
```

```
xfmt = mdates.DateFormatter('%d-%m-%y %H:%M')
ax.xaxis.set_major_locator(mdates.HourLocator(interval=3)) #to get a tick every 3 hours
ax.xaxis.set_major_formatter(xfmt)
```

#VALORES IMPORTANTES

```
maximo1=df4['salida1'].max()
minimo1 =df4['salida1'].min()
media1=df4['salida1'].mean()
```

```
maximo2=df4['salida2'].max()
minimo2=df4['salida2'].min()
media2=df4['salida2'].mean()
```

```
maximo3=df4['salida3'].max()
minimo3 =df4['salida3'].min()
media3=df4['salida3'].mean()
```

```
maximo4=df4['salida4'].max()
minimo4 =df4['salida4'].min()
media4=df4['salida4'].mean()
```

```
maximo5=df4['salida5'].max()
minimo5 =df4['salida5'].min()
media5=df4['salida5'].mean()
```

```
maximo6=df4['salida6'].max()
minimo6 =df4['salida6'].min()
media6=df4['salida6'].mean()
```

```
maximo7=df4['salida7'].max()
minimo7 =df4['salida7'].min()
media7=df4['salida7'].mean()
```

```
maximo8=df4['salida8'].max()
minimo8 =df4['salida8'].min()
media8=df4['salida8'].mean()
```

#Determinar el max/min/media global (toda la regleta)

```
list1=[maximo1,maximo2,maximo3,maximo4,maximo5,maximo6,maximo7,maximo8]
list2=[minimo1,minimo2,minimo3,minimo4,maximo5,minimo6,minimo7,minimo8]
list3=[media1,media2,media3,media4,media5,media6,media7,media8]
```

```
def maximo(list):
    maxnum=0
    for num in list:
        if num > maxnum:
            maxnum=num
        else:
            maxnum=maxnum
    return maxnum
```

```
def minimo(list):
    minnum=list[0]
    for num in list:
        if num< minnum:
            minnum=num
        else:
            minnum=minnum
    return minnum
```

```
def media(list):
    suma=0
    for num in list:
        suma=suma+num

    return suma/len(list)
```

#CREAR ALARMA

#DataFrame de la alarma 1

```
alarmas4=df4[(df4['salida1'] > 125)]
alarmas4['value']=df4['salida1']
alarmas4['salida']='1'
```

```
alarmas5=df4[(df4['salida2'] > 125)]
alarmas5['value']=df4['salida2']
alarmas5['salida']='2'
```

```
alarmas6=df4[(df4['salida3'] > 125)]
alarmas6['value']=df4['salida3']
alarmas6['salida']='3'
```

```
alarmas7=df4[(df4['salida4'] > 125)]
alarmas7['value']=df4['salida4']
alarmas7['salida']='4'
```

```
alarmas8=df4[(df4['salida5'] > 125)]
alarmas8['value']=df4['salida5']
alarmas8['salida']='5'
```

```
alarmas9=df4[(df4['salida6'] > 125)]
alarmas9['value']=df4['salida6']
alarmas9['salida']='6'
```

```
alarmas10=df4[(df4['salida7'] > 125)]
alarmas10['value']=df4['salida7']
alarmas10['salida']='7'
```

```
alarmas11=df4[(df4['salida8'] > 125)]
alarmas11['value']=df4['salida8']
alarmas11['salida']='8'
```

#Juntamos DataFrames

```
alarmase=alarmas4.append(alarmas5, ignore_index=True)
alarmase=alarmase.append(alarmas6, ignore_index=True)
alarmase=alarmase.append(alarmas7, ignore_index=True)
alarmase=alarmase.append(alarmas8, ignore_index=True)
alarmase=alarmase.append(alarmas9, ignore_index=True)
alarmase=alarmase.append(alarmas10, ignore_index=True)
alarmase=alarmase.append(alarmas11, ignore_index=True)
```

```
alarmase['tiposensor']= 'mPower'
alarmase['event']= 'Voltaje H'
```

#DataFrame de la alarma 2

```
#alarmas2=df4[(10 < df4['salida1'] < 110) | (10 < df4['salida2'] < 110) | (10 < df4['salida3'] <
110) | (10 < df4['salida4'] < 110) | (10 < df4['salida5'] < 110) | (10 < df4['salida6'] < 110) |
(10 < df4['salida7'] < 110) | (10 < df4['salida8'] < 110)]
alarmas12=df4[(df4['salida1'] < 110)]
alarmas12['value']=df4['salida1']
alarmas12['salida']='1'
```

```
alarmas13=df4[(df4['salida2'] < 110)]
alarmas13['value']=df4['salida2']
```

```
alarmas13['salida']='2'
```

```
alarmas14=df4[(df4['salida3'] < 110)]  
alarmas14['value']=df4['salida3']  
alarmas14['salida']='3'
```

```
alarmas15=df4[(df4['salida4'] < 110)]  
alarmas15['value']=df4['salida4']  
alarmas15['salida']='4'
```

```
alarmas16=df4[(df4['salida5'] < 110)]  
alarmas16['value']=df4['salida5']  
alarmas16['salida']='5'
```

```
alarmas17=df4[(df4['salida6'] < 110)]  
alarmas17['value']=df4['salida6']  
alarmas17['salida']='6'
```

```
alarmas18=df4[(df4['salida7'] < 110)]  
alarmas18['value']=df4['salida7']  
alarmas18['salida']='7'
```

```
alarmas19=df4[(df4['salida8'] < 110)]  
alarmas19['value']=df4['salida8']  
alarmas19['salida']='8'
```

#Juntamos DataFrames

```
alarmase2=alarmas12.append(alarmas13, ignore_index=True)  
alarmase2=alarmase2.append(alarmas14, ignore_index=True)  
alarmase2=alarmase2.append(alarmas15, ignore_index=True)  
alarmase2=alarmase2.append(alarmas16, ignore_index=True)  
alarmase2=alarmase2.append(alarmas17, ignore_index=True)  
alarmase2=alarmase2.append(alarmas18, ignore_index=True)  
alarmase2=alarmase2.append(alarmas19, ignore_index=True)
```

```
alarmase2['tiposensor']= 'mPower'  
alarmase2['event']= 'Voltaje L'
```

#Juntamos DataFrames en uno solo

```
alarmase3=alarmase.append(alarmase2, ignore_index=True)  
print alarmase3
```

```
del alarmase3['salida1']  
del alarmase3['salida2']  
del alarmase3['salida3']
```

```
del alarmase3['salida4']
del alarmase3['salida5']
del alarmase3['salida6']
del alarmase3['salida7']
del alarmase3['salida8']
```

#Ordenamos las alarmas por tiempo

```
alarmase3=alarmase3.sort_values(by='time')
```

```
plt.close()
```

V. Movimiento (movimiento.py)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import time
import datetime
import matplotlib.dates as mdates
import math
import openpyxl
from pandas import ExcelWriter
```

```
doc4=pd.read_csv('Motion&Door.csv')
df5=pd.DataFrame(doc4,columns=['row','last_val_time','sensor','event'])
df5.rename(columns={'last_val_time':'time'}, inplace=True)
```

#EPOCH TO HUMAN

```
df5['time']=df5['time']/1000
df5['time'] = df5['time'].apply(lambda x: datetime.datetime.fromtimestamp(x))
```

#COMPLETAR CAMPOS VACIOS (con False ya que la probabilidad de False respecto True es mas alta)

```
df5['event']=df5['event'].fillna(False)
```

#Identificar sensor

```
df5['sensor'] = df5['sensor'].apply(lambda x: 'Sensor movimiento' if x == 1 else 'Sensor puerta')
```

```
move=df5[df5['sensor'] == 'Sensor movimiento']
door=df5[df5['sensor'] == 'Sensor puerta']
```

#Recuento de true's que se han dado (ha ocurrido el evento)

```
eventmove=np.sum(move['event'])
eventdoor=np.sum(door['event'])
```

#Dividimos ventana para tener dos graficos

```
plt.subplot(1,2,1)
plt.pie([eventmove, len(move)-eventmove], labels = ['si','no'])
plt.title(u'% de veces que se detecto movimiento')
```

```
plt.subplot(1,2,2)
plt.pie([eventdoor, len(door)-eventdoor], labels=['si','no'])
plt.title(u'% de veces que se abrio la puerta')
```

#CREAR ALARMA**#DataFrame de la alarma 1**

```
alarmasm=df5[df5['event'] == True]
alarmasm['tiposensor']= df5['sensor']
alarmasm['event']= np.where(alarmasm['tiposensor'] == 'Sensor movimiento', 'Hay
movimiento', 'Abre puerta')
alarmasm['value']= df5['event']
del alarmasm['sensor']
```

#Ordenamos las alarmas por tiempo

```
alarmasm=alarmasm.sort_values(by='time')
```

```
plt.close()
```

VI. Sensor corriente (CurrentSensor.py)

```
import numpy as np
import pandas as pd
import datetime
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import time
```

```
df = pd.read_csv('mFimminute5.csv')
```

#CREATE DATAFRAME

```
df1 = pd.DataFrame(df,columns=['time', 'size', 'sld', 'max', 'min', 'ave', 'last_val',
'last_val_time'])
```

#CONVERSION EPOCH TO HUMAN TIME

```
df1['last_val_time']=df1['last_val_time']/1000
```

```
df1['last_val_time'] = df1['last_val_time'].apply(lambda x:
datetime.datetime.fromtimestamp(x))
df1['time']=df1['time']/1000
df1['time'] = df1['time'].apply(lambda x: datetime.datetime.fromtimestamp(x))
```

#IDENTIFICAR SENSOR 1 o 2

```
df1['sld'] = df1['sld'].apply(lambda x: 'Sensor1' if x ==
'ObjectId(59de90453004f138372d638f)' else 'Sensor2')
```

```
maximo=df1['max'].max()
minimo=df1['min'].min()
media=df1['last_val'].mean()
```

```
s1= df1[df1['sld'] == 'Sensor1']
s2= df1[df1['sld'] == 'Sensor2']
```

#GRAFICAR SENSORES

```
fig, ax = plt.subplots()
fig.autofmt_xdate() #Para adaptar indice x a formato fecha
```

```
plt.xlabel('Fecha y hora')
plt.ylabel('Corriente [amps]')
plt.title('Sensores corriente')
plt.plot(s1['time'], s1['ave'])
plt.plot(s2['time'], s2['ave'])
```

```
xfmt = mdates.DateFormatter('%d-%m-%y %H:%M')
ax.xaxis.set_major_locator(mdates.HourLocator(interval=3)) #to get a tick every 3 hours
ax.xaxis.set_major_formatter(xfmt)
```

```
plt.close()
```

VII. Eventos y alarmas (events.py)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import time
import datetime
import matplotlib.dates as mdates
import random
import math

from tempint import alarmast
```

```

from humedad import alarmash
from EnergiaElectrica import alarmase3
from movimiento import alarmasm

from EventsSensors import df1
alarmasc=pd.DataFrame()
alarmasc['event']=df1['rulename']
alarmasc['row']=df1.index
alarmasc['salida']= ''
alarmasc['time']=df1['time']
alarmasc['tiposensor']=df1['sensor']
alarmasc['value']=df1['sensor_value']

TOT=alarmast.append(alarmash, ignore_index=True)
TOT=TOT.append(alarmase3, ignore_index=True)
TOT=TOT.append(alarmasm, ignore_index=True)
TOT=TOT.append(alarmasc, ignore_index=True)
TOT=TOT.sort_values(by='time')

```

#Importar a excel

```

writer = pd.ExcelWriter('eventsexport.xlsx')
TOT.to_excel(writer,'Sheet1')
writer.save()

plt.close()

```

VIII. Relaciones (relaciones.py)

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from tempext import df
from CurrentSensors import df1
from tempint import df2
from humedad import df3
from EnergiaElectrica import df4

#RELACION CORRIENTE Y TEMPINT
corriente=df1.ix[:,5]
long= len(df1['ave'])-1
temp=df2['value'].ix[:long]

```



```
print np.corrcoef(corriente,temp)

plt.subplot(1,2,1)
plt.scatter(corriente,temp)
plt.title('Corriente y temperatura interior')
plt.ylabel('Tint')
plt.xlabel('Corriente')
plt.axis([0, 20, -20,70])
```

#RELACION TEMPEXT Y TEMPINT

```
tempexte=df.ix[:,2]
long1=len(tempexte)-1
tempint=df2['value'].ix[:long1]

print np.corrcoef(tempexte,tempint)

plt.subplot(1,2,2)
plt.scatter(tempexte,tempint)
plt.title('Text y Tint')
plt.ylabel('Tint')
plt.xlabel('Text')
plt.axis([-10,50,-20,70])

plt.close()
```

C. Ficheros .xlsx creados en eventos y alarmas

I. Fichero eventsexport.xlsx (a partir de datasets de Mockaroo)

Se muestran las primeras 40 filas de un total de 2050.

	event	row	salida	time	tiposensor	value
0	Humedad L	755		2017-10-11 16:42:00	Humedad	24,42
217	Voltaje L	755	7	2017-10-11 16:42:00	mPower	102,11
216	Voltaje L	755	1	2017-10-11 16:42:00	mPower	109,46
215	Voltaje H	1	4	2017-10-11 16:42:00	mPower	129,63
214	Voltaje L	755	2	2017-10-11 16:42:00	mPower	108,94
213	Voltaje H	755	4	2017-10-11 16:42:00	mPower	126,54

222	Voltaje H	756	4	2017-10-11 16:43:00	mPower	125,76
221	Voltaje H	756	6	2017-10-11 16:43:00	mPower	125,77
220	Voltaje L	756	1	2017-10-11 16:43:00	mPower	109,67
219	Voltaje L	2	4	2017-10-11 16:43:00	mPower	100,64
218	Voltaje L	2	3	2017-10-11 16:43:00	mPower	109,67
1580	Hay movimiento	756		2017-10-11 16:43:00	Sensor movimiento	1
1581	Hay movimiento	757		2017-10-11 16:44:00	Sensor movimiento	1
223	Voltaje L	3	3	2017-10-11 16:44:00	mPower	108,47
1	Humedad H	3		2017-10-11 16:44:00	Humedad	57,48
228	Voltaje H	4	5	2017-10-11 16:45:00	mPower	125,01
1582	Abre puerta	758		2017-10-11 16:45:00	Sensor puerta	1
227	Voltaje L	4	6	2017-10-11 16:45:00	mPower	107,58
226	Voltaje L	4	3	2017-10-11 16:45:00	mPower	109,3
225	Voltaje L	4	8	2017-10-11 16:45:00	mPower	107,73
224	Voltaje L	758	1	2017-10-11 16:45:00	mPower	109,43
2	Humedad L	758		2017-10-11 16:45:00	Humedad	25,45
229	Voltaje L	759	1	2017-10-11 16:46:00	mPower	107,21
230	Voltaje L	5	8	2017-10-11 16:46:00	mPower	107,73
3	Humedad L	6		2017-10-11 16:47:00	Humedad	26,46
4	Humedad H	760		2017-10-11 16:47:00	Humedad	57,13
1583	Abre puerta	760		2017-10-11 16:47:00	Sensor puerta	1
1584	Abre puerta	6		2017-10-11 16:47:00	Sensor puerta	1
231	Voltaje L	6	2	2017-10-11 16:47:00	mPower	108,7
234	Voltaje L	7	3	2017-10-11 16:48:00	mPower	108,34
232	Voltaje L	7	2	2017-10-11 16:48:00	mPower	108,71

233	Voltaje H	7	5	2017-10-11 16:48:00	mPower	125,31
238	Voltaje L	8	1	2017-10-11 16:49:00	mPower	105,67
239	Voltaje L	762	1	2017-10-11 16:49:00	mPower	108,55
237	Voltaje H	8	4	2017-10-11 16:49:00	mPower	127,17
235	Voltaje L	762	3	2017-10-11 16:49:00	mPower	108,3
5	Humedad L	762		2017-10-11 16:49:00	Humedad	15,23
236	Voltaje H	762	7	2017-10-11 16:49:00	mPower	126,14
1585	Hay movimiento	9		2017-10-11 16:50:00	Sensor movimiento	1

II. Fichero EventsSensorexport.xlsx (creado a partir del mFi Controller Software)

	time	rulename	sensor	sensor_value
45	2017-10-17 18:05:21	Corriente L	Sensor Corriente 2	0,6543
48	2017-10-17 18:35:26	Corriente L	Sensor Corriente 2	0,4021
53	2017-10-17 23:32:12	Corriente H	Sensor Corriente 1	0,0163
56	2017-10-17 23:41:55	Corriente L	Sensor Corriente 2	0,4645